Approximation Algorithms for Union and Intersection Covering Problems

Marek Cygan\textsuperscript{1}, Fabrizio Grandoni\textsuperscript{2}, Stefano Leonardi\textsuperscript{3}, Marcin Mucha\textsuperscript{1}, Marcin Pilipczuk\textsuperscript{1}, and Piotr Sankowski\textsuperscript{1}

\textsuperscript{1} Institute of Informatics, University of Warsaw, Poland
\{cygan, mucha, malcin, sank\}@mimuw.edu.pl
\textsuperscript{2} University of Rome Tor Vergata, Roma, Italy, grandoni@disp.uniroma2.it
\textsuperscript{3} Department of Computer and System Science, Sapienza University of Rome, Italy, leon@dis.uniroma1.it

Abstract In a classical covering problem, we are given a set of requests that we need to satisfy (fully or partially), by buying a subset of items at minimum cost. For example, in the \(k\)-MST problem we want to find the cheapest tree spanning at least \(k\) nodes of an edge-weighted graph. Here nodes and edges represent requests and items, respectively.

In this paper, we initiate the study of a new family of multi-layer covering problems. Each such problem consists of a collection of \(h\) distinct instances of a standard covering problem (layers), with the constraint that all layers share the same set of requests. We identify two main sub-families of these problems:

- in a union multi-layer problem, a request is satisfied if it is satisfied in at least one layer;
- in an intersection multi-layer problem, a request is satisfied if it is satisfied in all layers.

To see some natural applications, consider both generalizations of \(k\)-MST. Union \(k\)-MST can model a problem where we are asked to connect at least \(k\) users to either one of two communication networks, e.g., a wireless and a wired network. On the other hand, intersection \(k\)-MST can formalize the problem of providing both electricity and water to at least \(k\) users.

We present a number of hardness and approximation results for union and intersection versions of several standard optimization problems: MST, Steiner tree, set cover, facility location, TSP, and their partial covering variants.

1 Introduction

In the fundamental Minimum Spanning Tree problem (MST), the goal is to compute the cheapest tree which spans all the \(n\) nodes of a given edge-weighted graph \(G = (V, E)\). To handle the subtleties of real-life applications, several natural generalisations and variants of the problem have been considered.

For example, in the Steiner Tree problem we need to connect with a tree only a given subset \(W\) of \(k\) terminal nodes. In the \(k\)-MST problem instead, the
goal is to connect at least $k$ (arbitrary) nodes. One common feature of these generalizations is that we need to design a single network. However, this is often not the case in the applications. For example, suppose we want to provide at least $k$ out of $n$ users with both electricity and water. In this case, we cannot design the water and electricity infrastructures independently: our decisions on which users to reach have to be synchronized.

Consider now another classic problem, the Travelling Salesman problem (TSP): here we are given a complete weighted graph, and the goal is to compute the minimum-length tour traversing all the nodes. Again, several natural generalizations and variants of the problem have been considered in the literature. Still, all of them deal only with the case where there is a single network. However, there are natural applications which do not fit in this framework. For example, suppose you want to visit a set of places (bank, post office, etc.), and you can use your bike and your car. Of course, you cannot just reach a place by bike, and then suddenly switch to your car (that you left at home). Your trip must consist of a tour by bike and another tour by car, which together touch all the places that you need to visit.

The above examples show the need for a new framework, which is able to capture coordinated decision-making over multiple optimization problems.

Our results. In this paper we initiate the study of multi-layer covering problems. These problems are characterized by a set of $h$ instances of a standard covering problem (layers), sharing a common set of $n$ requests. The goal is satisfying, possibly partially, the requests by buying items in each layer at minimum total cost. We identify two main families of such problems:

- **Intersection problems.** Here, as in the water-electricity example, a request is satisfied if it is satisfied in all the layers.
- **Union problems.** Here, as in the car-bike example, a request is satisfied if it is satisfied in at least one layer.

We provide hardness and approximation results for the union and intersection versions of several classical covering problems: MST, Steiner Tree, (Non-Metric and Metric) Facility Location, TSP, and Set Cover. (Formal definitions are given at the end of this section). We focus on the partial covering variant of these problems, i.e. $k$-MST, $k$-Steiner Tree, etc.: here we need to satisfy a target number $k$ of the $n$ requests. This allows us to handle a wider spectrum of interesting problems. In fact, for intersection problems, if $k = n$ it is sufficient to compute an independent solution for each layer. On the other hand, some of the union problems above are interesting also for the case $k = n$. However, the results that we achieve for that case are qualitatively the same as for $k < n$.

For Intersection versions of $k$-MST, $k$-Steiner Tree, $k$-TSP, $k$-Set Cover, $k$-Metric Facility Location, and $k$-Nonmetric Facility Location, we show that:

- Even for two layers, a polylogarithmic approximation for these problems would imply a polylogarithmic approximation for $k$-Densest Subgraph.
We recall that the best approximation for the latter problem is $O(n^{1/4+\varepsilon})$ \cite{9} and finding a polylogarithmic approximation is a major open problem. Indeed, many researchers believe that a polylogarithmic approximation does not exist, and exploit this assumption in their hardness reductions (see, e.g., \cite{112}).

- On the positive side, we give $\tilde{O}(k^{1-1/h})$-approximation algorithms\footnote{The $\tilde{O}$ notation suppresses polylogarithmic factors.} for these problems.

Note that, in the single-layer case, the above problems can be approximated within a constant or logarithmic factor. Hence, our results show that the complexity of natural intersection problems changes drastically from one to two layers.

For Union versions of $k$-MST, $k$-Steiner Tree, $k$-TSP and $k$-Metric Facility Location we show that:

- The problems are $\Omega(\log k)$-hard to approximate for an unbounded number $h$ of layers. Furthermore, there is a greedy $O(\log k)$-approximation algorithm. For the first three problems this only holds for the rooted version — the unrooted case is inapproximable.
- There is an LP-based algorithmic framework which provides $O(h)$-approximate solutions. Furthermore, the natural LPs involved have $\Omega(h)$ integrality gap.

We remark that Union $k$-Set Cover and Union $k$-Nonmetric Facility Location can be solved by collapsing all layers into one, and hence they are less interesting with respect to the goals of this paper.

Related Work. To the best of our knowledge, and somewhat surprisingly, approximation algorithms for union and intersection problems seem to not have been studied in the literature, with the notable exception of Matroid Intersection. However, differently from our problems, Matroid Intersection is solvable in polynomial time \cite{12}.

The term “multi-layer” has been used before in the literature, but with a meaning different from ours. Most often it refers to problems related to VLSI design, where we are given several planar layers on which the circuit has to be built \cite{45}. It also sometimes refers to multi-layer models of communication networks that are composed of different physical and logical layers of communication devices \cite{55}.

The idea of introducing multiple cost functions into one optimization problem is the main theme of multi-objective optimization. Standard and multi-criteria approximation algorithms have been developed for the multi-objective version of several classical problems, such as Shortest Path \cite{27,33,34,43}, Spanning Tree \cite{20,22,34,87}, Matching \cite{6,7,20,34} etc. (for a survey, see \cite{13}). One could view these problems as having several layers with different costs. However, this setting is very different from our approach. In fact, solutions in different layers of multi-objective optimization problems have to be exactly the same, and the goal is to satisfy some constraints on each objective.
Partial covering problems (also known as problems with outliers), are well-studied in the literature: e.g., $k$-MST $[3,4,10,17,18,38]$, $k$-TSP $[3,17]$, $k$-Metric Facility Location $[11,29]$, and $k$-Set Cover $[30,40]$. Their generalization on multiple layers is significantly harder, as our results show. Note that our Union $k$-Steiner Tree problem generalizes all of the following problems: $k$-Steiner Tree (and hence $k$-MST), Prize-Capturing Steiner Tree (see the proof of Theorem 6), and $k$-Set Cover (see the proof of Theorem 7).

Rent-or-buy $[14,15,25,41]$ and buy-at-bulk $[19,21,24,25,42]$ problems can be seen as multi-layer problems where edge weights in different layers differ by a multiplicative factor. In contrast, weights of different layers are unrelated in our framework. Union $k$-TSP has some points in common with multi-depot versions of TSP $[32,36]$: also in that case multiple tours are computed; however, their weights are measured w.r.t. a unique weight function.

Recently Krishnaswamy et al. $[31]$ considered a matroid median problem, where a set of open centers must form an independent set from a matroid. This can be viewed as a generalisation of a Union problem, however in the matroid median problem all the centers are in the same metric space. This setting is less general than ours as it does not allow for modelling a setting with several completely unrelated metric spaces.

Preliminaries. In covering problems we are given a set $\mathcal{U}$ of $n$ requests, and a set $\mathcal{S}$ of items, with costs $w : \mathcal{S} \to \mathbb{R}_{\geq 0}$. The goal is to satisfy all requests by selecting a subset of items at minimum cost. We already defined MST, Steiner Tree, and TSP. Here, nodes and edges represent requests and items (with costs $w : E \to \mathbb{R}_{\geq 0}$, respectively. In the Set Cover problem, requests are the elements of a universe $\mathcal{U}$, and items $\mathcal{S}$ are subsets $S_1, \ldots, S_m$ of $\mathcal{U}$. Any $S_i$ satisfies all the $v \in S_i$. Nonmetric Facility Location is a generalization of Set Cover, where we are given a set $\mathcal{F}$ of facilities, with opening costs $o : \mathcal{F} \to \mathbb{R}_{\geq 0}$, and a set $\mathcal{C}$ of clients, with connection costs $w : \mathcal{C} \times \mathcal{F} \to \mathbb{R}_{\geq 0}$. The goal is to compute a subset $A$ of open facilities such that $\sum_{f \in A} o(f) + \sum_{c \in \mathcal{C}} w(c, A)$ is minimized. Here $w(c, A) := \min_{f \in A} w(c, f)$. We also say that $c$ is connected to (or served by) $A(c) := \arg\min_{f \in A} w(c, f)$. If connection costs satisfy triangle inequality, the problem is called Metric Facility Location.

We can naturally define partial covering versions for the above problems: $k$-MST, $k$-Steiner Tree, $k$-TSP, $k$-Nonmetric Facility Location, and $k$-Metric Facility Location.

It is straightforward to define union and intersection versions of the above problems. In the rest of this paper, the number of layers is denoted by $h$, and variables associated to layer $i$ have an apex $i$ (e.g., $w^i$, $o^i$, etc.), whereas $OPT$ denotes the optimum solution, and $opt$ its cost. By $N$ we denote the total number of requests and items (in all layers).

---

5 In the literature $k$-Nonmetric Facility Location often means that we are allowed to open at most $k$ facilities, while here we mean that we need to connect at least $k$ clients. Similarly for $k$-Metric Facility Location. Sometimes $k$-Set Cover indicates a Set Cover instance where the largest cardinality of a set is $k$, while our problem is sometimes called Partial Set Cover.
**Figure 1** Approximation algorithm for 2-layer INTERSECTION $k$-SET COVER.
For $a \in \{1, 2\}$, $\overline{a}$ is the other value in $\{1, 2\}$

1: procedure SCI($k, U, S^1, S^2, w^1, w^2$)
2: $K \leftarrow \emptyset$, $A^1 \leftarrow \emptyset$, $A^2 \leftarrow \emptyset$
3: repeat
4: for $a=1$ to 2 do
5: for all $X \in S^a$ do
6: for $b := 1$ to min($k - |K|$, $|X \setminus K|$) do
7: Solve one-layer INTERSECTION $k$-SET COVER problem on layer $\overline{a}$
8: with universe $X \setminus K$ and target $b$.
9: Let $(a', b', X')$ be the loop iterators which provide a solution $(K', A')$
10: minimizing the ratio of cost $C'$ to number $b'$ of covered elements.
11: $K \leftarrow K \cup K'$, $A^a \leftarrow A^a \cup \{X'\}$, $A^a' \leftarrow A^a' \cup A'$
12: until $|K| = k$
13: return $(K, A^1, A^2)$

By standard reductions, a $\rho$-approximation for the $k$-MST problem implies a $2\rho$-approximation for $k$-STEINER Tree and $k$-TSP. Moreover, a $\rho$-approximation for $k$-TSP gives a $2\rho$-approximation for $k$-MST. Essentially, the same reductions extend to the union and intersection versions of these problems. For this reason, in the rest of this paper we will consider the union and intersection version of $k$-MST only.

## 2 Intersection Problems

### 2.1 INTERSECTION $k$-SET COVER

In this section we present our approximation algorithm for INTERSECTION $k$-SET COVER. We recall that in this problem we are given $h$ collections $S^1, S^2, \ldots, S^h$ of subsets of a given universe $U$, where $w^i : S^i \rightarrow \mathbb{R}_{\geq 0}$ is the cost of subsets in the $i$th collection. The goal is covering at least $k$ elements in all layers simultaneously, at minimum total cost.

The basic idea behind our algorithm is as follows. We consider any set $X$ in any layer, and any number $j \leq k$ of elements in $X$. We solve recursively, on the remaining layers, the intersection problem induced by $X$ with target $j$. The base of the induction is obtained by solving a one-layer INTERSECTION $k$-SET COVER problem, using the greedy algorithm which provides a $(1 + \ln k)$-approximation [40]. We choose the set $X$ and the cardinality $j$ for which we obtain the best ratio of cost to number of covered elements. Next, we include covered elements in the solution under construction, and the problem is reduced consequently.

In order to highlight the main ideas of our approach, we focus on the special case $h = 2$, and we neglect polylogarithmic factors in the analysis. It is easy, just more technical, to extend the same approach to $h > 2$ and to refine (slightly) the approximation factor (See Appendix A.3).
**Theorem 1.** There is a $\tilde{O}(\sqrt{k})$-approximation algorithm for Intersection $k$-Set Cover on two layers.

**Proof.** Consider the algorithm in Figure 1. Its running time is polynomial, since SCI procedure calls the one-layer greedy algorithm $O(Nk^2)$ times.

Let $(O^1, O^2) \subseteq S^1 \times S^2$ be the optimal solution, and let $K_O \subseteq (\cup_{S \in O^1} S) \cap (\cup_{S \in O^2} S)$ be any set of $k$ elements in the intersection. For each element $x \in K_O$ and layer $i = 1, 2$, let us fix a set $O^i(x) \in O^i$ that covers $x$. We prove that at each iteration of the main loop $C'/b' = \tilde{O}(\text{opt}/\sqrt{k-|K|})$. This implies that the total cost of the constructed solution is bounded by $\sum_{i=0}^{k-1} \tilde{O}(\text{opt}/\sqrt{k-i}) = \text{opt} \cdot \tilde{O}(\sqrt{k})$.

Let $\kappa := \sqrt{k-|K|}$. We consider two cases, depending on whether there exists a set $X$ in the optimal solution that covers at least $\kappa$ elements of $K_O \setminus K$.

**Case 1.** Assume that there exists $1 \leq a \leq 2$ and $X \in O^a$, such that for at least $\kappa$ elements $x$ of $K_O \setminus K$ we have $O^a(x) = X$. Let us focus on the moment when our algorithm considers taking the set $X$. Obviously we have $\kappa \leq k - |K|$, therefore our algorithm considers covering $b := \kappa$ elements of $X$. As the optimal solution does it, it may be done with cost $\text{opt}$, so the call to the one layer algorithm returns a solution with cost $\tilde{O}(\text{opt})$. Hence we have $C'/b' = O(\text{opt}/\sqrt{k-|K|})$.

**Case 2.** For each $1 \leq a \leq 2$ and every $X \in O^a$, at most $\kappa$ elements of $K_O \setminus K$ satisfy $O^a(x) = X$. For each $x \in K_O \setminus K$, let $w(x) := w^1(O^1(x)) + w^2(O^2(x))$ be the sum of the costs of sets covering $x$ in the optimal solution. We have

$$\sum_{x \in K_O \setminus K} \sum_{a=1}^{2} \sum_{X \in O^a} \sum_{x \in K_O \setminus K : O^a(x) = X} w^a(O^a(x)) \leq \sum_{a=1}^{2} \sum_{X \in O^a} w^a(X) \kappa \leq \kappa \cdot \text{opt}.$$

Thus there exists $x_0 \in K_O \setminus K$ such that $w(x_0) \leq \kappa \cdot \text{opt}/|K_O \setminus K|$. If we take any $a$ and consider the iteration with $X = O^a(x_0)$ and $b = 1$, the algorithm computes a set of minimum cost $C_0 \leq w(x_0)$ covering $x_0$. We can conclude that

$$\frac{C'}{b'} \leq C_0 \leq \frac{\kappa \cdot \text{opt}}{|K_O \setminus K|} = \tilde{O}(\text{opt}/\sqrt{k-|K|}).$$

The proof of the following theorem is in Appendix A.3 due to space limits.

**Theorem 2.** There exists a $(4k^{1-1/h} \log^{1/h}(k))$-approximation algorithm for Intersection $k$-Nonmetric Facility Location (hence for Intersection $k$-Set Cover) running in $N^{O(h)}$ time.

### 2.2 Intersection $k$-MST

In this section we present a simple approximation algorithm for Intersection $k$-MST. Recall that here we are given a graph $G = (V, E)$ on $n$ nodes, and $h$ edge-weight functions $w^1, \ldots, w^h$. By taking the metric closures of $w^i$ we may
assume that \( G \) is complete. The goal is computing a tree \( T^i \) for each layer such that \( \sum_i w^i(T^i) \) is minimized and \( |\bigcap_i V(T^i)| \geq k \).

The algorithm is very simple: We consider a new metric \( w \) defined as a sum \( w(e) := \sum_i w^i(e) \) for each \( e \in E \), and compute a 2-approximate solution of the resulting (one-layer) \( k \)-MST problem using the algorithm in [18].

In Appendix A.1 we prove the following theorem.

**Theorem 3.** The Intersect \( k \)-MST algorithm above is \( 16k^{\frac{1}{2}} \)-approximate.

The analysis of the approximation ratio of the above algorithm given in Theorem 3 is tight up to a factor \( O(h) \) (see Appendix A.1).

### 2.3 Approximation Hardness

This section is devoted to the approximation hardness of Intersect \( k \)-MST, Intersect \( k \)-Set Cover (hence also of Intersect \( k \)-Nonmetric Facility Location) and Intersect \( k \)-Metric Facility Location. We use reductions from the \( k \)-Densest Subgraph problem: find the induced subgraph on \( k \) nodes with the largest possible number of edges. The fact that partial coverage problems can be as hard as \( k \)-Densest Subgraph is already known. Hajiaghayi and Jain [26] use \( k \)-Densest Subgraph to show that a partial coverage version of the Steiner Forest problem has no polylogarithmic approximation. In particular they introduce the Minimum \( \ell \)-Edge Coverage problem where one is to find the minimum number of vertices in a graph, whose induced subgraph has at least \( \ell \) edges. Moreover Hajiaghayi and Jain show a relation between approximation ratios for \( k \)-Densest Subgraph and Minimum \( \ell \)-Edge Coverage. In order to simplify our reductions we extend the result on Minimum \( \ell \)-Edge Coverage to bipartite graphs and prove the following theorems in Appendix A.2.

**Theorem 4.** If there exists an \( f(n) \)-approximation algorithm for unweighted Intersect \( k \)-Set Cover on two layers or for Intersect \( k \)-Metric Facility Location on two layers, then there exists a \( 16(f(2m))^2 \)-approximation algorithm for \( k \)-Densest Subgraph.

**Theorem 5.** If there exists an \( f(n) \)-approximation algorithm for Intersect \( k \)-MST on two layers, then there exists a \( 16(f(2n+2m+2))^2 \)-approximation algorithm for \( k \)-Densest Subgraph.

Theorems 4 and 5 suggest that the existence of a polylogarithmic approximation algorithm for the considered problems is rather unlikely (or at least very hard to achieve).

### 3 Union Problems

In this section we present our results for Union \( k \)-MST and Union \( k \)-Metric Facility Location. For Union \( k \)-MST, we first consider the rooted case in Sections 3.1 and 3.2 and then the unrooted one in Section 3.3. The Union MST problem is a variant of Union \( k \)-MST with \( k = n \).
3.1 Approximation Hardness

**Theorem 6.** Rooted Union $k$-MST and Union $k$-Metric Facility Location are APX-hard for any $h \geq 1$. Union MST is APX-hard for any $h \geq 2$.

*Proof.* The first claim trivially follows from the APX-hardness [18,23] of the considered problems for $h = 1$, by adding dummy layers with infinite edge weights.

For the second claim, we consider a reduction from the APX-hard [8] Prize-Collecting Steiner Tree problem: given an undirected graph $G = (V, E)$, edge weights $w : E \to \mathbb{R}_{\geq 0}$, a root node $r \in V$, and node prizes $p : V \to \mathbb{R}_{\geq 0}$, find a tree $T \ni r$ which minimizes $\sum_{e \in T} w(e) + \sum_{v \in T} p(v)$. We create a first layer, with edge weights $w_1 = w$. Then we construct a second layer, where we set $w_2(\{r, v\}) = p(v)$ for any $v \in V$. All the other layers, if any, are dummy layers defined as above. This reduction is approximation preserving. \hfill \Box

For an unbounded number of layers, our problems become much harder.

**Theorem 7.** For an arbitrary number of layers, rooted Union $k$-MST and Union $k$-Metric Facility Location are not approximable better than $\Omega(\log k)$ unless $P = NP$, even when $k = n$.

*Proof.* We prove the claim for rooted Union $k$-MST, by giving a reduction from cardinality Set Cover: given a universe $U$ of $n'$ elements, and a collection $S = \{S_1, \ldots, S_m\}$ of $m'$ subsets of $U$, find a minimum cardinality subset $A \subseteq S$ which spans $U$. This problem is $\Omega(\log n')$-hard to approximate [39]. We create one node per element of $U$, plus two extra nodes $r$ and $s$. We create one layer $i$ for each set $S_i$ (i.e., $h = m'$). In layer $i$ we let $w_1(\{r, s\}) = 1$ and $w_1(\{s, v\}) = 0$ for each $v \in S_i$. We also let $r^i := r$ for each $i$, and assume $k = n = n' + 2$. Note that any solution to the rooted Union $k$-MST instance of cost $\alpha$ can be turned into a solution to the Set Cover instance of the same cost, and vice versa.

To prove the claim for Union $k$-Metric Facility Location, we use the same reduction as above, where the edge $\{r, s\}$ is replaced by a single node $r$, which is a facility of opening cost 1. \hfill \Box

In Appendix B.1 we give a greedy $O(\log k)$-approximation algorithm.

3.2 An LP-Based Approximation for rooted Union $k$-MST

In this section we present an LP-based $O(\log h)$-approximation algorithm for rooted Union $k$-MST. Essentially the same approach works also for Union $k$-Metric Facility Location (see Appendix B.2). This is an improvement over the $\Theta(\log k)$-approximation given by the greedy algorithm for the relevant case of bounded $h$.

For notational convenience, we assume that the roots $R := \cup_i \{r^i\}$ are not counted into the target number $k$ of connected nodes. In other terms, we replace $k$ by $k - |R|$. We make the same assumption also in the case of one layer. Consider
the following LP relaxation for $k$-Steiner Tree ($W \ni r$ is the set of terminals) denoted by $LP_{kST}(w, W, V, r, k)$:

$$\min \sum_{e \in E} w(e) x_e$$

s.t.

$$\sum_{e \in \delta(S)} x_e \geq z_v, \quad \forall (v, S) : S \subseteq V - \{r\}, v \in S \cap W;$$

$$\sum_{v \in W} z_v \geq k;$$

$$x_e \geq 0, 1 \geq z_e \geq 0, \quad \forall v \in W, \forall e \in E.$$

Here, variable $x_e$ indicates whether edge $e$ is included in the solution, whereas variable $z_v$ indicates whether terminal $v$ is connected. Moreover $\delta(S)$ denotes the set of edges with exactly one endpoint in $S$. Observe that $LP_{kMST}(w, V, r, k) := LP_{kST}(w, V, r, k)$ is an LP relaxation for $k$-MST. We need the following lemmas.

**Lemma 8.** \cite{77} Let $(w, V, r, k)$ be an instance of $k$-MST, $w_{\text{max}} := \max_{v \in V} w(r, v)$, and $\text{opt}'$ be the optimal solution to $LP_{kMST}(w, V, r, k)$. There is a polynomial-time algorithm $\text{apx-kmst}$ which computes a solution to the instance of cost at most $2\text{opt}' + w_{\text{max}}$.

**Lemma 9.** \cite{628} Let $G = (V \cup \{v\}, E)$ be a directed graph, with edge capacities $\alpha : E \to \mathbb{R}_{\geq 0}$ such that $\sum_{e \in \delta^+(u)} \alpha(e) = \sum_{e \in \delta^-(u)} \alpha(e)$ for all $u \in V \cup \{v\}$. Then there is a pair of edges $(u, v)$ and $(v, z)$, such that the following capacity reservation $\beta$ supports the same flow as $\alpha$ between any pair of nodes in $V$: for $\Delta \alpha := \min \{\alpha(u, v), \alpha(v, z)\}$, set $\beta(u, v) = \alpha(u, v) - \Delta \alpha$, $\beta(v, z) = \alpha(v, z) - \Delta \alpha$, $\beta(u, z) = \alpha(u, z) + \Delta \alpha$, and $\beta(e) = \alpha(e)$ for the remaining edges $e$.

**Corollary 10.** Given a feasible solution $(x, z)$ to $LP_{kST}(w, W, V, r, k)$, there is a feasible solution $(x', z')$ to $LP_{kMST}(w, W, r, k)$ such that $\sum_e w(e)x'_e \leq 2 \cdot \sum_e w(e)x_e$.

**Proof.** Variables $x_e$ can be interpreted as a capacity reservation which supports a fractional flow of value $z_e$ from each $v \in W$ to the root. Let us replace each edge with two oppositely directed edges, and assign to each such edge the same weight and capacity as the original edge. This way, we obtain a capacity reservation $\alpha$ which costs twice the original capacity reservation, and satisfies the condition of Lemma 9. We consider any non-terminal node $v \neq r$ with some incident edge of positive capacity, and apply Lemma 9 to it. Due to triangle inequality, the cost of the capacity reservation does not increase. We iterate the process on the resulting capacity reservation. Within a polynomial number of steps, we obtain a capacity reservation $\beta$ which: (1) supports the same flow from each terminal to the root $r$ as $\alpha$, (2) has value 0 on edges incident to non-terminal nodes (besides $r$), and (3) does not cost more than $\alpha$. At this point, we remove the nodes $V - (W \cup \{r\})$, and merge the capacity of oppositely directed edges to get an undirected capacity reservation $x'$. By construction, the pair $(x', z)$ is a feasible solution to $LP_{kMST}(w, W, r, k)$ of cost at most $2 \cdot \sum_e w(e)x_e$. \qed
We are now ready to describe our algorithm for rooted Union $k$-MST. In a preliminary step we guess the largest distance $L$ in the optimal solution between any connected node and the corresponding root, and discard nodes at distance larger than $L$ from their root. This introduces a factor $O(nh)$ in the running time. Note that $L \leq \text{opt}$.

We let $V^i$ be the remaining nodes in layer $i$.

Then we compute the optimal fractional solution $\text{OPT}^* = (x^i, z^i, s^i)$, of cost $\text{opt}^*$, to the following LP relaxation $L_P_{ukMST}$ for the problem, where variables $x^i_e$ and $z^i_v$ indicate whether edge $e$ is included in the solution of layer $i$ and node $v$ is connected in layer $i$, respectively. Variable $z_v$ indicates whether node $v$ is connected in at least one layer.

\[
\begin{align*}
\min \quad & \sum_{i=1,\ldots,h} \sum_{e \in E} w^i(e) x^i_e \\
\text{s.t.} \quad & \sum_{e \in \delta(S)} x^i_e \geq z^i_v, \quad \forall i \in \{1,\ldots,h\}, \forall (v,S) : S \subseteq V^i - \{r^i\}, v \in S; \\
& \sum_{i=1,\ldots,h} z^i_v \geq z_v, \quad \forall v \in V - R; \\
& \sum_{v \in V - R} z_v \geq k; \\
& z^i_v, x^i_e \geq 0, 1 \geq z_v \geq 0, \quad \forall i \in \{1,\ldots,h\}, \forall v \in V - R, \forall e \in E.
\end{align*}
\]

Given $\text{OPT}^*$, we compute for each layer $i$ a subset of nodes $W^i$, where $v$ belongs to $W^i$ if $z^i_v = \max_{j=1,\ldots,h} \{z^j_v\}$ (breaking ties arbitrarily). We also define $k^i := \lceil \sum_{v \in W^i} z_v \rceil$. For each layer $i$, we consider the $k$-MST instance on nodes $W^i \cup \{r^i\}$ with target $k^i$. This instance is solved using the 2-approximation algorithm $\text{apx-kmst}$ of Lemma 8; the resulting tree $T^i$ is added to the solution for layer $i$. Let $k'$ be the number of connected nodes. If $k' < k$, the algorithm connects $k - k'$ extra nodes, chosen greedily, to the corresponding root in order to reach the global target $k$.

**Theorem 11.** There is a $O(h)$-approximation algorithm for rooted Union $k$-MST. The running time of the algorithm is $O((nh)^{O(1)})$.

**Proof.** Consider the algorithm above. The claim on the running time is trivial. By construction, the solution computed is feasible (i.e., it connects $k$ nodes). It remains to consider the approximation factor.

For each $v \in W^i$, we let $\tilde{z}^i_v = z_v$, and set $\tilde{z}^i_v = 0$ for the remaining nodes. Furthermore, we let $\tilde{x}^i_e = h \cdot x^i_e$. Observe that $(\tilde{x}^i, \tilde{z}^i, z^i)$ is a feasible fractional solution to $L_P_{ukMST}$ of cost $h \cdot \text{opt}^*$. Observe also that $(\tilde{x}^i, \tilde{z}^i)$ is a feasible solution to $L_P_{kST}(w^i, W^i, V^i, \{r^i\})$; let $a\tilde{x}^i$ be the associated cost. By Lemma 8 there is a fractional solution to $L_P_{kST}(w^i, W^i, r^i, k^i)$ of cost at most $2a\tilde{x}^i$. It follows from Lemma 8 that the solution computed by $\text{apx-kmst}$ on layer $i$ costs at most $4a\tilde{x}^i + L$.

Since the $W^i$’s are disjoint, the algorithm initially connects at least $\sum_i k^i \geq k - h$ nodes. Hence the cost of the final augmentation phase is at most $h \cdot L \leq h \cdot \text{opt}$. Putting everything together, the cost of the solution returned by the algorithm is at most:

$$
\sum_i (4a\tilde{x}^i + L) + h \cdot L \leq 4h \cdot \text{opt}^* + 2h \cdot L \leq 6h \cdot \text{opt}.
$$
The constant multiplying $h$ in the approximation factor can be reduced with a more technical analysis, at the cost of a higher running time. We also observe that the integrality gap of $\mathcal{LP}_{akMST}$ is $\Omega(h)$ (see Appendix B.3).

### 3.3 Unrooted Union $k$-MST

**Theorem 12.** Unrooted Union $k$-MST is not approximable in polynomial time for an arbitrary number $h$ of layers unless $P=NP$.

**Proof.** We give a reduction from SAT: given a CNF boolean formula on $m'$ clauses and $n'$ variables, determine whether it is satisfiable or not. For each variable $i$, we create two nodes $t_i$ and $f_i$. Intuitively, these nodes represent the fact that $i$ is true or false, respectively. Furthermore, we have a node for each clause. Hence the overall number of nodes is $n = 2n' + m'$. We create a separate layer for each variable $i$ (i.e., $h = n'$). In layer $i$, we connect with an edge of cost zero $t_i$ (resp., $f_i$) to all the clauses which are satisfied by setting $i$ to true (resp., to false). The target value is $k = n' + m'$. Note that, there is a satisfying assignment to the SAT instance iff there is a solution of cost zero to the Union $k$-MST instance.

For $h = O(1)$, the rooted and unrooted versions of the problem are equivalent approximation-wise. In fact, one obtains an approximation-preserving reduction from the unrooted to the rooted case by guessing one node $r^i$ in the optimal solution per layer: this introduces a polynomial factor $O(n^h)$ in the running time. We remark that an exponential dependence on $h$ of the running time is unavoidable in the unrooted case, due to Theorem 12. An opposite reduction is obtained by appending $n$ dummy nodes to each root (distinct nodes for distinct layers), with edges of cost zero, and setting the target to $k + hn$. The following result follows.

**Corollary 13.** Unrooted Union $k$-MST is APX-hard for any $h \geq 1$. There is an $O(h)$-approximation algorithm for the problem of running time $O((hn)^{O(1)} n^h)$.

### 4 Conclusions and Open Problems

In this paper, we introduced multi-layer covering problems, a new framework that can be used to describe a wide spectrum of yet unstudied problems. We addressed two natural ways of combining the layers: intersection and union. We gave multi-layer approximation algorithms, as well as hardness results, for a few classic covering problems (and their partial covering versions). There are several research questions that merit further study.

- There are other natural ways one can combine the layers. Consider, for example, the car/bike problem in the case where you can put your bike in the car trunk. Now you can make more than one tour by bike, the only requirement being that the bike tours all touch the (unique) car tour.

---

6 Without loss of generality, we can assume that each clause does not contain both a literal and its negation.
– What about min-max multi-layer problems, where the goal is minimizing the maximum cost over the layers?
– We considered covering problems: what about packing problems?
– Our algorithms for union problems give tight bounds only with respect to the corresponding natural LPs. This leaves room for improvement.
– There is a considerable gap between upper and lower bounds for intersection problems. In particular, our hardness results do not depend on $h$, while the approximation ratios deteriorate rather rapidly for increasing $h$.

References

39. R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-
constant error-probability PCP characterization of NP. In STOC, pages 475–484, 1997.


41. C. Swamy and A. Kumar. Primal-dual algorithms for connected facility location


43. Z. Tarapata. Selected multicriteria shortest path problems: An analysis of com-


45. C.-H. Yeh, E. A. Varvarigos, and B. Parhami. Multilayer VLSI layout for inter-
A Intersection Problems

In this section we give the omitted details about the intersection problems.

A.1 Intersection k-MST

In this section we prove a simple approximation algorithm for Intersection k-MST. Recall that here we are given a graph $G = (V, E)$ on $n$ nodes, and $h$ edge-weight functions $w^1, \ldots, w^h$. By taking the metric closures of $w_i$ we may assume that $G$ is complete. The goal is computing a tree $T$ for each layer such that $\sum_i w^i(T^i)$ is minimized and $|\bigcap_i V(T^i)| \geq k$.

The algorithm is very simple: We consider a new metric $w$ defined as a sum $w(e) := \sum_i w^i(e)$ for each $e \in E$, and compute a 2-approximate solution of the resulting (one-layer) $k$-MST problem using the algorithm in [13].

**Lemma 14.** Let $K \subseteq V$, and $w^i(K)$ denote the cost of the minimum spanning tree of $K$ on layer $i$. Then there exist two nodes $u, v \in K$ such that $w^i(u, v) \leq 4w^i(K)/|K|^{1/h}$ for $i = 1, \ldots, h$.

**Proof.** Let us prove the following claim by induction on $i$: for any $i \in \{0, \ldots, h-1\}$, there exist a nodeset $K_i \subseteq K$ and paths $P_i^1, P_i^2, \ldots, P_i^i$ on $K_i$ such that:

1. $|K_i| \geq |K|^{1-i/h}$
2. $w^j(P_i^j) \leq 2w^i(K)/|K|^{1/h}$ for $j = 1, \ldots, i$.

Trivially $K_0 = K$ satisfies the claim, hence assume $i > 0$. Let $T_i$ be the minimum spanning tree of $K$ on layer $i$. Duplicate its edges, compute an Euler tour, and shortcut duplicated nodes. Let $C_i$ be the resulting cycle on $K$ of length at most $2w^i(K)$. Remove up to $|K|^{1/h}$ edges from $C_i$ so as to obtain $|K|^{1/h}$ segments of length at most $2w^i(K)/|K|^{1/h}$ each. Let $P$ be the segment maximizing the cardinality of $K_i := V(P) \cap K_{i-1}$. Set $K_i$ satisfies (a) since $|K_i| \geq |K_{i-1}|/|K|^{1/h} \geq |K|^{1-(i-1)/h-1/h}$.

The paths $P_i^j$ and $P_i^j$, $j < i$, satisfying (b) are obtained from $P$ and $P_i^{i-1}$, respectively, by shortcutting the nodes not in $K_i$.

Similarly as above, we can split $C_i$ into $|K|^{1/h}/2$ segments which span $K$ and have length at most $4w^h(K)/|K|^{1/h}$ each. At least one of these segments contains $2|K_{h-1}|/|K|^{1/h} \geq 2$ nodes of $K_{h-1}$. Thus there are two nodes $u$ and $v$ such that $w^i(u, v) \leq 4w^i(K)/|K|^{1/h}$ for $i = 1, \ldots, h$. \hfill $\square$

**Theorem 15 (Theorem 3 restated).** The Intersection $k$-MST algorithm above is $16k^{1-1/h}$-approximate.

**Proof.** Consider the following process: starting with the optimal set $K_{\ominus}$ of $k$ covered nodes, we iteratively take the edge $\{x, y\}$ guaranteed by Lemma 14 and contract it in all layers, until $K_{\ominus}$ collapses into a single node. The contracted edges form a tree $T'$ (same for all layers) spanning $k$ nodes, of cost

$$w(T') \leq 4 \sum_{i=1}^h w^i(K_{\ominus}) \sum_{j=1}^{k-1} (k-i+1)^{-\frac{1}{h}} \leq 4 \sum_{i=1}^h w^i(K_{\ominus}) \int_1^k x^{-\frac{1}{h}} dx$$

$$< 8k^{1-\frac{1}{h}} \sum_{i=1}^h w^i(K_{\ominus}) = 8k^{1-\frac{1}{h}} \text{opt.}$$
The algorithm returns a solution of cost at most $2w(T')$. The claim follows. \qed

We show that the bound proven in Section 2 is tight up to a factor $O(h)$.

**Lemma 16.** The approximation ratio of the Intersection k-MST algorithm in Section 2 is $\Omega\left(\frac{\log n}{h}\right)$

**Proof.** Take an arbitrary integer $N > 2$ and set $n = 2^{hN} - 1$. We are going to construct weights $w^1, w^2, \ldots, w^h$ on an $n$-node complete graph $G = (V, E)$ such that $w^i(T^i) = n - 1$, but $w(T) = \Omega(n^{2-1/h})$. Here $T^i$ is the minimum spanning tree on layer $i$.

We take $V = \{0, 1, \ldots, n - 1\}$, that is, the set of nodes of $G$ are all numbers with up to $hN$ digits in binary, except for $2^{hN} - 1$, i.e., the number with $hN$ ones in binary. Given $x \in V$, we split its $hN$-digit binary representation into $h$ segments of length $N$ and denote the $i$-th segment by $x_i$. In other words, the binary representation of $x$ is $x_1x_2\ldots x_h$ and each $x_i$ is an $N$-digit binary string. By $x(i)$ we denote number represented in binary as $x_i x_{i+1} \ldots x_h x_1 \ldots x_{i-1}$, that is, we rotate cyclically $(i - 1)N$ bits.

To construct metric $w^i$, sort $V$ according to numbers $x(i)$ and connect $V$ into Hamiltonian cycle $C_i$ in this order. All edges on $C_i$ have weight 1 and other distances are minimum length distances on $C_i$. Clearly, $w^i(T^i) = n - 1$.

It is sufficient to show that, for each edge $x, y$, we have $w(x, y) = \Omega(n^{1-1/h})$. This leads to the claimed bound on $w(T)$. We distinguish a few subcases.

**Case 1.** There exists $i, 1 \leq i \leq h$, such that $|x_i - y_i| \geq 2$ and $\{x_i, y_i\} \neq \{0, 2^{N} - 1\}$. Then in $G_i$ the distance between $x$ and $y$ is at least $n^{1-1/h}$.

**Case 2.** There exists $i, 1 \leq i \leq h$, such that $x_i = y_i$. Take $j$ such that $x_j \neq y_j$, but $x_{j+1} = y_{j+1}$ (with $x_{h+1} = x_1$). Then in $G_j$ the distance between $x$ and $y$ is at least $n^{1-1/h} - n^{1-2/h}$.

**Case 3.** There exists $i, 1 \leq i \leq h$, such that $|x_i - y_i| = 1$. Then in $G_i$ (with $G_0 = G_h$) the distance between $x$ and $y$ is at least $n^{1-1/h} - 2n^{1-2/h}$.

**Case 4.** As $2^{hN} - 1$ is not in $V$, there exists $i, 1 \leq i \leq h$, such that $x_i = 2^{N} - 1$, $y_i = 0$, but $x_{i+1} = 0$ and $y_{i+1} = 2^{N} - 1$. Then the distance between $x$ and $y$ in $G_i$ is at least $n^{1-1/h} - 2n^{1-2/h}$.

As we exhausted all possibilities, the bound on $w(T)$ is proven. \qed

**A.2 Approximation hardness**

We start with the following technical lemma.

**Lemma 17.** Assume we have an undirected graph $G = (V, E)$ and an induced subgraph $G[X]$, $X \subseteq V$ with $x$ nodes and $y$ edges. Let $2 \leq x_0 \leq x$ and let $c := x/x_0$. Then one can in polynomial time find an induced subgraph $G[Y]$ on $x_0$ nodes with at least $y/(2c^2)$ edges.

**Proof.** By the linearity of expectations a random subset $X_0 \subseteq X$ containing $x_0$ vertices induces a subgraph $G[X_0]$ with $\frac{xx_0(x_0-1)}{2(x-1)} \geq \frac{y^2}{2c^2} = \frac{xy_0}{2c^2}$ edges. We can derandomize this procedure with standard techniques. \qed
Now we reduce the domain of $k$-DENSEST SUBGRAPH to bipartite graphs (see also Figure 2).

**Lemma 18.** Assume there exists a $f(n,k)$-approximation algorithm for $k$-DENSEST SUBGRAPH on bipartite graphs. Then there exists a $8f(2n,2k)$-approximation algorithm for the same problem on arbitrary graphs.

**Proof.** Assume we have an instance of $k$-DENSEST SUBGRAPH, i.e., a graph $G = (V, E)$ and one integer $k$. Construct a bipartite graph $G' = (V_1 \cup V_2, E')$ as follows: for each $v \in V$ we take two copies $v_1 \in V_1$ and $v_2 \in V_2$. For each $(u, v) \in E$ we add $(u_1, v_2)$ and $(u_2, v_1)$ to $E'$. Let us run the $f(n,k)$-approximation algorithm for the bipartite graph $G'$ and $k' := 2k$. This way we obtain a set $X' \subset V_1 \cup V_2$ such that $G[X']$ has $y'$ edges. Take $X := \{v : v_1 \in X' \text{ or } v_2 \in X'\}$, $k \leq |X| \leq 2k$. The graph $G[X]$ has at least $y'/2$ edges. Reduce $X$ to size $k$ using Lemma 17 obtaining a solution with at least $y'/16$ edges.

Let us now bound how much the obtained solution is worse than the optimal solution. Let $X_{opt}$ be any optimal solution in $G$ such that $G[X_{opt}]$ has $y_{opt}$ edges and $k$ nodes. In $G'$ set $X'_{opt} = \{v_1, v_2 : v \in X_{opt}\}$ has $2k$ nodes and $2y_{opt}$ edges, thus $y' \geq 2y_{opt}/f(2n,2k)$. Therefore $G[X]$ has at least $y_{opt}/(8f(2n,2k))$ edges.

Now we relate $k$-DENSEST SUBGRAPH and MINIMUM $\ell$-EDGE COVERAGE. A lemma similar to the following one was proved in [26], but we include the proof for the sake of completeness.

**Lemma 19.** Assume there exists a $f(n)$-approximation algorithm for MINIMUM $\ell$-EDGE COVERAGE on bipartite graphs. Then there exists a $2(f(n))^2$-approximation algorithm for $k$-DENSEST SUBGRAPH on bipartite graphs.

**Proof.** Assume we have an instance $(G = (V, E), k)$ of $k$-DENSEST SUBGRAPH. For the graph $G$ we run the approximation algorithm for MINIMUM $\ell$-EDGE COVERAGE with consecutive $\ell := 1, 2, \ldots$, obtaining solutions $X_1, X_2, \ldots$. We stop when $G[X_{\ell+1}]$ has more that $f(n)k$ nodes. Assume that $X_\ell$ was the last solution with at most $f(n)k$ nodes. We reduce $X_\ell$ to size $k$ using Lemma 17 and return the reduced set.
Let us now prove that it is in fact a $2(f(n))^2$-approximation. Let $X_{opt}$ be the optimal solution for the $k$-DENSEST SUBGRAPH instance with $y_{opt}$ edges. Note that $\ell \geq y_{opt}$, as $X_\ell$ was the last solution with at most $f(n)k$ nodes and our algorithm is a $f(n)$-approximation. Thus, by Lemma 17, the returned solution has at least $\ell/(2(|X_\ell|/k)^2) \geq y_{opt}/(2(f(n))^2)$ edges.

Pipelining Lemmas 19 and 18 proves the following theorem.

**Theorem 20.** If there exists an $f(n)$-approximation algorithm for MINIMUM $\ell$-EDGE COVERAGE on bipartite graphs, then there exists a $16(f(2n))^2$-approximation algorithm for $k$-DENSEST SUBGRAPH on arbitrary graphs.

We conclude this section by showing the missing reductions for INTERSECTION $k$-SET COVER, INTERSECTION $k$-MST and INTERSECTION $k$-METRIC FACILITY LOCATION.

**Lemma 21.** If there exists an $f(n,k)$-approximation algorithm for unweighted INTERSECTION $k$-SET COVER on two layers, then there exists an $f(m,\ell)$-approximation algorithm for MINIMUM $\ell$-EDGE COVERAGE on bipartite graphs.

**Proof.** Let $(G, \ell)$, $G = (V_1 \cup V_2, E)$, be the considered instance of MINIMUM $\ell$-EDGE COVERAGE. For each $v \in V_1 \cup V_2$, let $\delta(v)$ be the set of edges incident to $v$. Consider the 2-layer INTERSECTION $k$-SET COVER instance $(U, k, S^1, S^2)$ with: $U = E$, $k = \ell$, and $S^i = \{\delta(v) : v \in V_i\}$ for $i = 1, 2$. Note that a solution for $(G, \ell)$ translates to a solution for $(U, k, S^1, S^2)$ and vice versa. □

**Lemma 22.** If there exists an $f(n,k)$-approximation algorithm for INTERSECTION $k$-METRIC FACILITY LOCATION on two layers, then there exists an $f(m,\ell)$-approximation algorithm for MINIMUM $\ell$-EDGE COVERAGE on bipartite graphs.

**Proof.** Let $(G, \ell)$, $G = (V_1 \cup V_2, E)$, be the considered instance of MINIMUM $\ell$-EDGE COVERAGE. For each $v \in V_1 \cup V_2$, let $\delta(v)$ be the set of edges incident to $v$. Consider the 2-layer INTERSECTION $k$-METRIC FACILITY LOCATION instance defined as follows. Let $C = E$, $k = \ell$ and $F^i = V_i$. We define all opening costs to be equal to 1 and all connection costs $w^i(e, v)$ to be equal to 0 if $v$ is an endpoint of $e$, or $\infty$ otherwise. As each client (edge) $e$ is connected by a finite distance to only one facility in each layer, costs $w^i$ are metric. Note that a solution for $(G, \ell)$ translates to a solution for $(C, k, F^1, F^2)$ and vice versa. □

**Lemma 23.** If there exists an $f(n,k)$-approximation algorithm for INTERSECTION $k$-MST on two layers, then there exists an $f(n+m+1,\ell)$-approximation algorithm for MINIMUM $\ell$-EDGE COVERAGE on bipartite graphs.

**Proof.** Let $(G, \ell)$, $G = (V_1 \cup V_2, E)$, be a bipartite instance of MINIMUM $\ell$-EDGE COVERAGE. We show how to construct the first layer of the corresponding INTERSECTION $k$-MST instance: the construction of the second layer is symmetric.

Consider the following auxiliary weighted graph $G' = (V', E')$. The nodeset $V'$ is given by $\{r\} \cup V_1 \cup V_2 \cup E$, where $r$ is a newly created root node. Moreover,
\( E' = E'_a \cup E'_b \cup E'_c \), where: \( E'_a = \{ \{ r, v \} : v \in V_2 \} \), \( E'_b = \{ \{ v, u \} : v \in V_1, \{ v, u \} \in E \} \), and \( E'_c \) to \( \infty \), 1 and 0, respectively. Intuitively, we want all nodes of \( G' \) that correspond to \( V_2 \) to be too expensive to be used in this layer. Eventually, we consider the metric closure of \( G' \) (set \( E' \) induces a tree). We set the target \( k := \ell \).

Because of the \( \infty \) edges, no node \( v \in V_1 \cup V_2 \) will belong to the intersection. Thus the only nodes in the intersection will be the nodes corresponding to edges of the original bipartite graph. Consequently a solution for Intersection \( k \)-MST of cost \( \alpha \) in \( G' \) translates to a solution for the Minimum \( \ell \)-Edge Cover instance with \( \alpha \) vertices and vice versa.

Pipelining each of the Lemmas 21, 22 and 23 with Theorem 20 we prove Theorems 4 and 5.

### A.3 Intersection \( k \)-Nonmetric Facility Location

In this section we give a \((4k^{1-1/h} \log^{1/h}(k))\)-approximation algorithm for Intersection \( k \)-Nonmetric Facility Location. The algorithm works in \( N^{O(h)} \) time, i.e., a polynomial time for any fixed \( h \).

Let us define the problem in a way which is convenient for our purposes. For each layer \( i \), we are given a collection \( F^i \) of subsets of the set \( C \) of clients, with one subset \( X \) for each facility \( f(X) \). Intuitively, \( X \) is the set of clients that facility \( f(X) \) is allowed to serve on layer \( i \). We use \( o^i(X) \) and \( w^i(x, X) \) as shortcuts for \( o^i(f(X)) \) and \( w^i(x, f(X)) \), respectively. A solution consists of a subset \( K \subseteq C \) of \( k \) clients, and a subset \( A^i \subseteq F^i \). Intuitively, \( K \) is the subset of clients that we decide to connect, and \( \cup_{X \in A^i} f(X) \) is the set of facilities that we open on layer \( i \) to connect \( K \) on that layer. We denote as \( A^i(x) \) the facility serving client \( x \) on layer \( i \). We slightly generalize Intersection \( k \)-Nonmetric Facility Location by adding for each \( x \in C \) a cost \( \overline{p}(x) \) of using \( x \) in the solution. In total, the cost of a solution is:

\[
\sum_{x \in K} \overline{p}(x) + \sum_{i=1}^{h} \sum_{X \in A^i} o^i(X) + \sum_{x \in K} \sum_{i=1}^{h} w^i \left( x, A^i(x) \right).
\]

As we shall see, this generalization does not make the problem much harder, but allows us to describe our algorithm more neatly.

A \( O(\log k) \)-approximation for our generalized Intersection \( k \)-Nonmetric Facility Location problem for \( h = 1 \) follows easily from [40].

**Lemma 24.** There exists a \((1 + \ln k)\)-approximation algorithm for the generalised \( k \)-Nonmetric Facility Location problem, where connecting client \( x \in C \) has extra cost \( \overline{p}(x) \).

**Proof.** We can reduce the generalised version of the \( k \)-Nonmetric Facility Location problem to the classical one by simply increasing all distances between client \( x \) and any facility \( f \) by \( \overline{p}(x) \). The claim follows from [40].
Figure 3 Approximation algorithm for Intersection k-Nonmetric Facility Location.

1: procedure FLI \((k, \mathcal{C}, h, (F_i)_i, (o, w, \overline{r}))\)
2: if \(h = 1\) then
3: return solution found by Lemma 24
4: if \(k = 1\) then
5: return optimal solution by brute-force
6: \(K \leftarrow \emptyset, A^i \leftarrow \emptyset\) for \(i = 1, 2, \ldots, h\).
7: repeat
8: for all \(1 \leq r \leq h\) and all \(X \in F^r\) do
9: for \(j = 1\) to \(\min(k - |K|, |X \setminus K|)\) do
10: \(\overline{r} \leftarrow \overline{r}, \text{except for elements } x \in X \setminus K, \text{where we put } \overline{r}(x) = \overline{r}(x) + o((x, X)).\)
11: \((K_{j,x}, (A^j_{i,x})_i) \leftarrow \text{FLI}(j, X \setminus K, h - 1, (F^i)_{1 \leq i \leq h, i \neq r}, (o, w, \overline{r})).\)
12: \(C_{j,x} \leftarrow \text{cost of } (K_{j,x}, (A^j_{i,x})_i) \text{ w.r.t. costs } (o, w, \overline{r}), \text{ plus } o'(X)\).
13: \(c_{j,x} \leftarrow C_{j,x} / j.\)
14: \(r_0, X_0, j_0 \leftarrow \text{values of the loops’ } \overline{r} \text{ for which the cheapest solution was found according to weights } c_{j,x}.\)
15: \(K \leftarrow K \cup K_{j_0,X_0}, A^0 \leftarrow A^0 \cup \{X_0\}, A^i \leftarrow A^i \cup A^i_{j_0,X_0} \text{ for } i \neq r_0\)
16: For each \(x \in K_{j_0,X_0} \) assign \(A^0(x) = X_0.\)
17: until \(|K| = k\)
18: return \((K, (A^i)_i)\)

Now we solve the problem for arbitrary \(h\). Let us give some notation. We are going to develop a procedure FLI \((k, \mathcal{C}, h, (F^i)_i, (o, w, \overline{r}))\) that returns the \((4k^{1-1/h} \log^{1/h}(k))-\text{approximation}\) for Intersection \(k\)-Nonmetric Facility Location. We abandon the requirement that each \(X \in \bigcup_i F^i\) is contained in \(\mathcal{C}\), but the solution set \(K\) needs to be a subset of \(\mathcal{C}\).

Note that Lemma 24 provides an algorithm for \(h = 1\). For arbitrary \(h\), we use the procedure described in Figure 3. The FLI procedure constructs set \(K\) with facilities \(A^i\) and a choice function greedily. At one step, we iterate over all layers \(r\) and sets \(X \in F^r\) and all possible cardinalities \(j\) of elements that set \(X\) can cover (i.e., \(1 \leq j \leq \min(k - |K|, |X \setminus K|)\)) and try to: choose \(X\) and solve problem without sets \(F^r\) (where \(X \in F^r\)) for universe restricted to \(X \setminus K\) and the goal size \(j\). We solve the subproblem using FLI procedure, but for \(h\) decreased by one. We hide the costs of attaching elements \(x \in X \setminus K\) to \(X\) in costs per element, i.e., in \(\overline{r}(x)\). Finally, we choose set \(X\) and cardinality \(j\) with the smallest cost per element covered. This cost is kept in variable \(c_{j,x}\).

**Proof. (Theorem 2)** We consider the algorithm in Figure 3. It is clear that FLI procedure works in \(NO(h)\) time, as it calls recursively itself \(O(Nk^2)\) times with \(h\) decreased by one.

We next prove the claim on the approximation by induction on \(h\). For \(h = 1\) the thesis is implied by Lemma 24. Assume now that all recursive calls in the FLI procedure return solutions with \((4k^{1-1/(h-1)} \log^{1/(h-1)}(k))\) approximation ratio.
Let \( opt \) be the cost of the optimal solution for the given instance. Pick any optimal solution with cost \( opt \) and let \( K_O \) be the set of covered elements by it, and \( O' \) be the chosen subset of \( F \) for \( i = 1, 2, \ldots, h \). For each layer \( i \) and each element \( x \in K_O \) we fix the set \( O^i (x) \in O^i \) that covers \( x \) in the optimal solution.

We prove that at one step of the algorithm, the weight \( c_{jo,X_o} \) satisfies:

\[
    c_{jo,X_o} \leq opt \cdot 4^{1-1/h}(k - |K|)^{-1/h} \log^{1/h}(k).
\]

Recall that \( c_{jo,X_o} \) is the average cost paid newly covered elements. This bound is sufficient, since the total cost of the constructed solution is bounded by:

\[
    \text{total cost} \leq opt \cdot \sum_{i=0}^{k-1} 4^{1-1/h}(k - i)^{-1/h} \log^{1/h}(k) \leq opt \cdot 4^{1-1/h} \log^{1/h}(k) \int_0^k x^{-1/h} dx
\]

\[
    = opt \cdot 4^{1-1/h} \log^{1/h}(k) \frac{1}{1 - 1/h} k^{1-1/h} \leq opt \cdot 4 \log^{1/h}(k) k^{1-1/h}.
\]

The last inequality follows from the fact that \( 4^{-\varepsilon} \leq 1 - \varepsilon \) for \( 0 \leq \varepsilon \leq \frac{1}{4} \).

Let \( \kappa := 4^{1-1/h}(k - |K|)^{1-1/h} \log^{1/h}(k) \). We consider two cases, depending on whether there exists a layer \( r \) and a set \( X \in O' \) that covers at least \( \kappa \) elements of \( K_O \setminus K \), i.e,

\[
    |\{x \in K_O \setminus K : O' (x) = X\}| \geq \kappa.
\]

**Case 1.** Assume there exists a layer \( r \) and a set \( X \in O' \) such that for at least \( \kappa \) elements \( x \) of \( K_O \setminus K \) we have \( O' (x) = X \). Let us focus on the moment when our algorithm considers taking set \( X \). We may assume \( \kappa \leq k - |K| \), as otherwise \( k - |K| \) is bounded by constant and we may instead use brute force to finish the greedy construction optimally. Therefore our algorithm considers covering \( \kappa \) elements of \( X \). As the optimal solution does it, it may be done with cost \( opt \), so the recursive call returns the solution with cost at most \( opt \cdot 4^{k-1/(h-1)} \log^{1/(h-1)}(\kappa) \). We cover \( \kappa \) elements, so

\[
    c_{\kappa,X} \leq opt \cdot 4^{k-1/(h-1)} \log^{1/(h-1)}(\kappa)
\]

\[
    \leq opt \cdot 4 \left( \frac{1}{h-1} (k - |K|) \frac{1}{h} \log^{1/h}(k) \right)^{\frac{1}{h-1}} \log^{1/(h-1)}(k)
\]

\[
    = opt \cdot 4^{1-1/h}(k - |K|)^{-1/h} \log^{1/h}(k).
\]

**Case 2.** Every \( x \in F_{opt} \) covers at most \( \kappa \) elements of \( K_O \setminus K \). For each \( x \in K_O \setminus K \) denote

\[
    \overline{w}(x) = \overline{w}(x) + \sum_{r=1}^h w^r(x, O^r (x)) + o^r (O^r (x)),
\]
i.e., the total cost of choosing \( x \), attaching it to set \( \mathcal{O}^r(x) \) and choosing set \( \mathcal{O}^r(x) \). By the assumption in this case, we have

\[
\sum_{x \in K \setminus \mathcal{O}} \overline{w}(x) = \sum_{x \in K \setminus \mathcal{O}} \overline{\sigma}(x) + \sum_{r=1}^{h} w^r(x, \mathcal{O}^r(x)) + \sum_{r=1}^{h} \sum_{X \in \mathcal{O}^r} \sigma^r(X) \cdot |\{x \in K \setminus \mathcal{O} : \mathcal{O}^r(x) = X\}| \leq \kappa \cdot \text{opt}.
\]

Thus there exists \( x_0 \in K \setminus \mathcal{O} \) such that \( \overline{w}(x_0) \leq \kappa \cdot \text{opt}/|K \setminus \mathcal{O}|. \) Let \( X_0 = \mathcal{O}^r(x_0) \). Note that, since our algorithm uses brute force for \( k = 1 \), the recursive call with find optimal solution for \( j := 1 \) and \( X := X_0 \) and thus \( c_{1,x_0} \leq \overline{w}(x_0) \).

As \( |K \setminus \mathcal{O}| \geq k - |K| \) we have:

\[
c_{1,x_0} \leq \kappa \cdot \text{opt}/|K \setminus \mathcal{O}|
\leq \text{opt} \cdot 4^{1-1/h}(k - |K|)^{1-1/h} \log^{1/h}(k - |K|)^{-1}
= \text{opt} \cdot 4^{1-1/h}(k - |K|)^{-1} \log^{1/h}(k).
\]

\[\square\]

**B Union Problems**

In this section we give the omitted details concerning union covering problems.

**B.1 A Greedy Approach**

We next describe a simple greedy algorithm which provides a logarithmic approximation for several union partial covering problems. Consider a partial covering problem where \( \mathcal{U} \) is the set of requests, \( \mathcal{S}^i \) is the set of items on layer \( i \), with costs \( w^i : \mathcal{S}^i \rightarrow \mathbb{R}_{\geq 0} \), and \( k \) is the target. We require that the covering problem satisfies a natural composition property, namely two solutions satisfying \( k' \) and \( k'' \) distinct requests, can be merged (without increasing the total cost) to obtain a solution satisfying \( k' + k'' \) requests. (Merging might involve some polynomial-time operations). The algorithm works as follows:

1. For all layers \( i \), for all \( k^i := 1, \ldots, k \), solve the single-layer problem induces by the triple \( (\mathcal{U}, \mathcal{S}^i, k^i) \) with a \( \rho \)-approximation algorithm.
2. Among all the solutions computed, take the one \( \mathcal{A} \), obtained for some triple \( (\mathcal{U}, \mathcal{S}^i, k^i) \), which minimizes the ratio \( w^i(\mathcal{A})/k^i \).
3. Merge \( \mathcal{A} \) with the solution under construction. Remove from \( \mathcal{U} \) the requests satisfied by \( \mathcal{A} \), and decrease \( k \) by \( k^i \).
4. If \( k > 0 \), go to Step (1). Otherwise return the current solution.

**Theorem 25.** The algorithm above computes a \( O(\rho \log k) \)-approximation for the partial covering problem considered in polynomial time.
Proof. The claim on the running time is trivial. The algorithm computes a feasible solution, due to the composition property. Consider now the approximation ratio. Let $A_1, \ldots, A_q$ be the sequence of approximate solutions computed, $w_j$ be the cost of $A_j$ and $k_j$ the number of requests that it satisfies on layer $i_j$. Observe that at the beginning of iteration $j$, the current number of requests is $k - \sum_{a<j} k_a$, and the cost of the optimal solution with respect to that number of requests is no more than $\text{opt}$. By an averaging argument, at each iteration $j$ we have $w_j = \rho \text{opt} / (k - \sum_{a<j} k_a)$. We can conclude that the cost of the solution computed is at most

$$\rho \text{opt} \left( \frac{k_1}{k} + \frac{k_2}{k-k_1} + \ldots + \frac{k_q}{k-\sum_{a<j} k_a} \right) \leq \rho \text{opt} \cdot \ln k.$$ 

\[\square\]

**Corollary 26.** There are $O(\log k)$-approximation algorithms for Union $k$-MST and Union $k$-Metric Facility Location.

Proof. Observe that removing requests transforms the original $k$-MST problem in each layer into a $k$-Steiner Tree problem: for the latter problem there is a 4-approximation algorithm [18]. Note also that all the partial solutions in layer $i$ contain the root $r^i$: hence the merging step is trivial. The claim for Union $k$-MST follows.

For $k$-Metric Facility Location, there is a 2-approximation algorithm in [29]. In this case removing a request simply means removing one client, and the merging step is trivial. This proves the claim for Union $k$-Metric Facility Location. \[\square\]

### B.2 Union $k$-Metric Facility Location

In this section we present an LP-based $O(h)$-approximation algorithm for Union $k$-Metric Facility Location. As we will see, the basic idea is an for Union $k$-MST.

Recall that in Union $k$-Metric Facility Location we are given a graph $G = (V, E)$, a set $C \subseteq V$ of clients, a set $F \subseteq V$ of facilities, one integer $k$ (target), a set of opening cost functions $o^i : F \rightarrow \mathbb{R}_{\geq 0}$, and a set of edge-weight functions $w^i : E \rightarrow \mathbb{R}_{\geq 0}$, with $i = 1, \ldots, h$. The distance between nodes $u$ and $v$ w.r.t. $w^i$ is denoted as $w^i(u, v)$. A feasible solution is given by a pair $(\tilde{C}^i, \tilde{F}^i)$ for each layer $i$, $\tilde{C}^i \subseteq C$ and $\tilde{F}^i \subseteq F$, such that $|\cup_i \tilde{C}^i| \geq k$. The goal is minimizing the cost $\sum_{i=1}^h (\sum_{f \in \tilde{F}^i} o^i(f) + \sum_{c \in \tilde{C}^i} w^i(c, \tilde{F}^i))$. Here $w^i(c, \tilde{F})$ denotes the minimum distance on layer $i$ between client $c \in C$ and facility $f \in F \subseteq F$. 
Also in this case we consider a natural LP relaxation $L_{\text{k-MFL}}(C, F, o, w, k)$ for the single-layer version of the problem:

$$\min \sum_{f \in F} o(f)y_f + \sum_{(c, f) \in C \times F} w(c, f)x_{c, f}$$

subject to:

- $x_{c, f} \leq y_f, \quad \forall (c, f) \in C \times F$;
- $\sum_{f \in F} x_{c, f} \geq z_c, \quad \forall c \in C$;
- $\sum_{c \in C} z_c \geq k$;
- $x_{c, f}, y_f \geq 0, 1 \geq z_c \geq 0, \quad \forall c \in C, \forall f \in F$.

Variable $y_f$ indicates whether facility $f$ is opened, and variable $x_{c, f}$ whether client $c$ is connected to facility $f$. Variable $z_c$ indicates whether client $c$ is connected to some facility. We need the following result.

**Lemma 27.** Let $(C, F, o, w, k)$ be an instance of $k$-Metric Facility Location, $o_{\text{max}} := \max_{f \in F} o(f)$, and $\text{opt}'$ be the optimal solution to $L_{\text{k-MFL}}(C, F, o, w, k)$. There is a polynomial time algorithm $\text{apx-kmfl}$ which computes a solution to the instance of cost at most $3\text{opt}' + 2o_{\text{max}}$.

The algorithm that we use is analogous to the one for the $k$-MST case. In a preliminary phase we guess the largest cost $o^*$ of a facility in the optimum solution, and remove all facilities of larger cost. Let $F^i$ be the remaining set of facilities on layer $i$. We then compute the optimal solution $OPT^* = (x^i, y^i, z^i, z^i)$, of cost $\text{opt}^*$, to the following relaxation $L_{\text{u-kMFL}}$ for the problem:

$$\min \sum_{i=1}^{h} \sum_{f \in F^i} o^i(f)y^i_f + \sum_{(c, f) \in C \times F^i} w^i(c, f)x^i_{c, f}$$

subject to:

- $x^i_{c, f} \leq y^i_f, \quad \forall i \in \{1, \ldots, h\}, \forall (c, f) \in C \times F^i$;
- $\sum_{f \in F^i} x^i_{c, f} \geq z^i_c, \quad \forall i \in \{1, \ldots, h\}, \forall c \in C$;
- $\sum_{i=1}^{h} z^i_c \geq k$;
- $x^i_{c, f}, y^i_f, z^i_c \geq 0, 1 \geq z_c \geq 0, \quad \forall i \in \{1, \ldots, h\}, \forall c \in C, \forall f \in F^i$.

Then we identify for each layer $i$ the subset of clients $C^i := \{c \in C : z^i_c = \max_j \{z^j_c\}\}$. We run the algorithm $\text{apx-kmfl}$ from Lemma 27 on each layer, with clients $C^i$, facilities $F^i$, and target $k^i := \sum_{c \in C^i} z^i_c$; we open facilities and connect clients accordingly. Let $k^i < k$, we connect extra clients in a greedy fashion, possibly opening new facilities: in particular, we consider the pairs $(c, f) \in C^i \times F^i$, with $c$ not connected, which minimize $o^i(f) + w^i(c, f)$, and we connect the corresponding clients.

**Theorem 28.** There is a $O(h)$-approximation algorithm for $\text{Union k-Metric Facility Location}$. The running time of the algorithm is $O((nh)^{O(1)})$.

**Proof.** Consider the algorithm above. The claim on the running time is trivial. As in the $k$-MST case, consider the feasible fractional solution $(\tilde{x}^i, \tilde{y}^i, \tilde{z}^i, z^i)$ obtained from $OPT^*$ by setting $\tilde{z}^i_c = z_c$ if $c \in C^i$, $\tilde{z}^i_c = 0$ otherwise, and raising the
variables $x$ and $y$ by a factor $h$. This new solution costs at most $h \cdot \text{opt}^*$. Furthermore, $(\tilde{x}^i, \tilde{y}^i, \tilde{z}^i)$ is a feasible fractional solution to $LP_{kMFL}(C^i, F^i, o^i, w^i, k^i)$. Let $a\tilde{x}^i$ be its cost. Lemma 27 guarantees that the cost of the integral solution on layer $i$ is at most $3a\tilde{x}^i + 2o^*$. Since $\sum_i k^i \geq k - h$, the final step costs at most $h \cdot \text{opt}$. Altogether the cost of the solution computed is at most

$$\sum_i (3a\tilde{x}^i + 2o^*) + h \cdot \text{opt} \leq 3h \cdot \text{opt}^* + 2h \cdot o^* + h \cdot \text{opt} \leq 6h \cdot \text{opt}.$$ 

Also in this case a more technical analysis allows one to reduce the constant in front of $h$ in the approximation factor, at the cost of a larger running time.

B.3 Integrality Gap

Lemma 29. The integrality gap of $LP_{ukMST}$ and $LP_{ukMFL}$ is $\Omega(h)$.

Proof. We consider the following unweighted SET COVER instance given in [44].

Let $G'$ be a hypergraph on $m'$ nodes, which has one hyperedge for any subset of $m'/2$ nodes. We construct a set cover instance with $m'$ sets given by nodes, \(\binom{m'}{m'/2}\) elements given by hyperedges, and inclusion given by incidence. Taking a fraction $2/m'$ of each set gives a feasible fractional solution of cost 2 to the natural set cover LP. On the other hand, the optimal integral solution uses $m'/2 + 1$ sets. Hence the integrality gap in this case is $\Omega(m')$.

The same reductions as in Theorem 7 imply a $\Omega(h)$ lower bound on the integrality gap of $LP_{ukMST}$ and $LP_{ukMFL}$ for the case $k = n$. \qed