Approximating Star Cover Problems

Buddhima Gamlath
École Polytechnique Fédérale de Lausanne, Switzerland
buddhima.gamlath@epfl.ch

Vadim Grinberg
Toyota Technological Institute at Chicago, Chicago, IL, USA
vgm@ttic.edu

Abstract

Given a metric space $(F \cup C, d)$, we consider star covers of C with balanced loads. A star is a pair (i, C_i) where $i \in F$ and $C_i \subseteq C$, and the load of a star is $\sum_{j \in C_i} d(i, j)$. In minimum load k-star cover problem (MLkSC), one tries to cover the set of clients C using k stars that minimize the maximum load of a star, and in minimum size star cover (MSSC) one aims to find the minimum number of stars of load at most T needed to cover C, where T is a given parameter.

We obtain new bicriteria approximations for the two problems using novel rounding algorithms for their standard LP relaxations. For MLkSC, we find a star cover with $(1 + O(\varepsilon))k$ stars and $O(1/\varepsilon^2)\mathrm{OPT}_{\mathrm{MLk}}$ load where $\mathrm{OPT}_{\mathrm{MLk}}$ is the optimum load. For MSSC, we find a star cover with $O(1/\varepsilon^2)\mathrm{OPT}_{\mathrm{MS}}$ stars of load at most $(2 + O(\varepsilon))T$ where $\mathrm{OPT}_{\mathrm{MS}}$ is the optimal number of stars for the problem. Previously, non-trivial bicriteria approximations were known only when $F = C$.

2012 ACM Subject Classification Theory of computation

Keywords and phrases star cover, approximation algorithms, lp rounding

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2020.57

Category APPROX

Related Version A full version of the paper is available at `https://arxiv.org/pdf/1912.01195`.

Acknowledgements We are very grateful to Ola Svensson for influential discussions at multiple stages of this work.

1 Introduction

Facility location (FL) is a family of problems in computer science where the general goal is to assign a set of clients to a set of facilities under various constraints and optimization criteria. FL family encompasses many natural clustering problems like k-median and k-means, most of which are well studied. In this work, we study two relatively less studied FL problems which we call minimum load k-star cover (MLkSC) and minimum size star cover (MSSC). The goal of MLkSC is to assign clients to at most k facilities, minimizing the maximum assignment cost of a facility, while that of MSSC is to find a client-facility assignment with the minimum number of facilities such that the total assignment cost of each facility is upper bounded by a given threshold T.

We begin by formally defining the two problems. Let C be a finite set of clients and F be a finite set of facilities. Let $(F \cup C, d)$ be a finite metric space where $d : (F \cup C) \times (F \cup C) \to \mathbb{R}_0^+$ is a distance metric. By a *star* in (F, C), we mean any tuple (i, C_i), where $i \in F$ and $C_i \subseteq C$. We say two stars (i, C_i) and (h, C_h) are *disjoint* if $i \neq h$ and $C_i \cap C_h = \varnothing$. A *star cover* of (F, C) is a finite collection $S = \{(i_1, C_{i_1}), \ldots, (i_{|S|}, C_{i_{|S|}})\}$ of disjoint stars such that $C = C_{i_1} \cup \cdots \cup C_{i_{|S|}}$. The size of a star cover S is the number of stars $|S|$ in the cover. Given a star cover S, a star $(i, C_i) \in S$, and a client $j \in C_i$, we say that client j is *assigned* to facility i under S and the facility i is *serving* client j under S. For a star (i, C_i), the *load* of facility i

is the sum of pair-wise distances $\sum_{j\in C_i} d(i,j)$ between itself and its clients. The *load* $L(S)$ of a star cover S is the load of its maximum load star. I.e., $L(S) := \max_{(i,C_i)\in S} \sum_{j\in C_i} d(i,j)$. For notational convenience, we denote the collection of all star covers of (F,C) by $\mathcal{S}$. Using the introduced notation, we now define MLkSC and MSSC.

▶ **Definition 1** (Minimum Load k-Star Cover)**.** *Given a finite metric space $(F \cup C, d)$ and number $k \in \mathbb{N}$, the task of minimum load k-star cover problem is to find a star cover of size at most k that minimizes the load; I.e., find $S^\star := \operatorname{argmin}_{S\in\mathcal{S}:|S|\le k} L(S)$. We denote the optimal load $L(S^\star)$ by $\mathrm{OPT}_{\mathrm{MLk}}$.*

▶ **Definition 2** (Minimum Size Star Cover)**.** *Given a finite metric space $(F \cup C, d)$ and a number $T \in \mathbb{R}_+$, the task of minimum size star cover problem is to find a star cover of load at most T that minimizes the size; I.e., find a star cover $S^\star := \operatorname{argmin}_{S\in\mathcal{S}:L(S)\le T} |S|$. We denote the optimal size $|S^\star|$ by $\mathrm{OPT}_{\mathrm{MS}}$.*

Even et al. [5] showed that both MLkSC and MSSC are NP-hard for general metrics even when $F = C$. Both Even et al. [5] and Arkin et al. [3] studied the problem in $F = C$ setting and gave constant factor bicriteria approximation algorithms for MLkSC. The latter work also gave a constant factor approximation algorithm for MSSC in the same setting.

Arkin et al. [3] use k-median clustering and then split the individual clusters that are too large into several smaller clusters to obtain their approximation guarantees. However, the splitting of clusters rely on that the clients and facilities are indistinguishable, which allows one to conveniently choose a new facility for each new partition created in the splitting process. Meanwhile, the technique of Even et al. [5] is to formulate the problem as an integer program, round its LP relaxation using minimum make-span rounding techniques, and use a clustering approach that also relies on F being equal to C to obtain the final bicriteria approximation guarantees. Both the techniques do not generalize to the case where $F \neq C$ unless it is allowed to open the same facility multiple times.

Recently, Ahmadian et al. [1] showed that MLkSC is NP-hard even if we restrict the metric space to be a line metric. They further gave a PTAS for MLkSC in line metrics and a quasi-PTAS for the same in tree metrics. However, their techniques are specific to line and tree metrics, and it is not known whether they can be extended to general metrics.

The main goal of this work is to extend the approach of Even et al. [5] to $F \neq C$ setting where any given facility can be opened at most once. To do so, we introduce a novel clustering technique and an accompanied new algorithm to modify the LP solution before applying the minimum makespan rounding at the end. This yields the following theorem:

▶ **Theorem 3.** *There exists a polynomial time algorithm that, given an instance $(F \cup C, d)$ of* MLkSC *problem and any $\varepsilon \in (0,1)$, finds a star cover of (F,C) of size at most $(1+O(\varepsilon))k$ and load at most $O(\mathrm{OPT}_{\mathrm{MLk}}/\varepsilon^2)$.*

As a complementary result, we also show that the standard LP relaxation has some inherent limitations. That is, we construct a family of MLkSC instances where the load of any integral $(1+\varepsilon)k$-star cover is at least $\Omega(1/\varepsilon)$ times the optimal value of the standard LP.

With slight modifications to our clustering and rounding techniques, we further obtain the following theorem on MSSC:

▶ **Theorem 4.** *There exists a polynomial time algorithm that, given an instance of* MSSC *problem with load parameter T and any $\varepsilon \in (0,1)$, finds a star cover of load at most $(2+O(\varepsilon))T$ and size at most $O(\mathrm{OPT}_{\mathrm{MS}}/\varepsilon^2)$.*

As with MLkSC, we show that the standard LP-relaxation for MSSC also suffers from inherent limitations; I.e., for any $\varepsilon > 0$, we give an instance of MSSC for which there is a *fractional* star cover of load at most T but any integral star cover of that instance has load at least $(2 - \varepsilon)T$ even with all facilities opened.

We end the introduction with a brief section on other related work. In Section 2, we introduce the LP relaxations of the two problems and provide a more elaborate description of our techniques. Later in Section 3 and Section 4 we describe the proofs of Theorem 3 and Theorem 4 in detail. We present the explicit constructions of families of MLkSC and MSSC that show inherent limitations of the respective standard LP relaxations in Appendix B.

Other Related Work

To the best of our knowledge, Even et al. [5] and Arkin et al. [3] were among the first to explicitly address close relatives of MLkSC and MSSC problems. Both of their works considered the problem where one has to cover nodes (or edges) of a graph using a collection of *objects* (I.e.,trees or stars). Even et al. considered the problem of minimizing the maximum cost of an object when the number of objects is fixed, for which they gave a 4-approximation algorithm. Arkin et al. also studied the same problem and additionally considered paths and walks as covering objects. They further discussed the MSSC version of the problems where the goal is to minimize the number of covering objects such that the cost of each object is at most a given threshold. For min-max tree cover with k trees, Khani and Salavatipour [6] later improved the approximation guarantee to a factor of three.

In general, many well-known facility location problems have constant factor approximation guarantees. For example, for uncapacitated facility location, the known best algorithm (Li et al. [8]) gives an approximation ratio of 1.488. For k-median in general metric spaces, the current best is 2.675 due to Byrka et al. [4], and for k-means in general metric spaces, it is $(9 + \varepsilon)$ due to Ahmadian et al. [2]. Remarkably, all these results follows from LP based approaches. A common theme of all these problems is that their objectives are to minimize a summation of costs. I.e., we minimize the sum of distances from clients to their respective closest opened facilities, where in uncapacitated facility location problem, we additionally have the sum of opening costs of the opened facilities. This *min-sum* style objective is in contrast with the min-max style objective of minimum star cover problem which makes it immune to algorithmic approaches that are applicable to other common facility location counterparts.

As discussed, minimum star cover problems are closely related to minimum makespan scheduling and the generalized assignment problem. Two most influential literature in this regard include Lenstra et al. [7] and Shmoys et al. [10].

2 Our Results and Techniques

We start with the LP relaxations of the standard integer program formulations for MLkSC and MSSC. To make the presentation easier, we first define a polytope SC-LP(T, k) such that the *integral* points of SC-LP(T, k) are feasible star covers of load at most T and size at most k.

For $i \in F$, let variable $y_i \in \{0, 1\}$ denote whether i'th facility is *opened* (I.e., $y_i = 1$ if and only if there is a star (i, C_i) in the target star cover), and for $(i, j) \in F \times C$, let variable $x_{ij} \in \{0, 1\}$ denote whether j'th client is *assigned* to facility i (I.e., $x_{ij} = 1$ if and only if $j \in C_i$ where (i, C_i) is a star in the target star cover). Then the following set of constraints define

SC-LP(T, k):

$$\begin{aligned}
&\sum_{j \in C} d(i,j) \cdot x_{ij} \le T \cdot y_i && \forall i \in F, && (1)\\
&\sum_{i \in F} y_i \le k, && && (2)\\
&\sum_{i \in F} x_{ij} = 1 && \forall j \in C, && (3)\\
&x_{ij} \le y_i && \forall i \in F, \forall j \in C, && (4)\\
&y_i \in [0,1] && \forall i \in F, && (5)\\
&x_{ij} \in [0,1] && \forall i \in F, \forall j \in C && (6)\\
&x_{ij} = 0 && \forall i \in F, \forall j \in C : d(i,j) > T. && (7)
\end{aligned} \qquad \text{(SC-LP}(T,k))$$

Here, Constraint (1) ensures that the load of an opened facility $i \in F$ is at most T, while Constraint (2) limits the maximum number of opened facilities to k. Constraint (3) and Constraint (4) ensure that each client is fully assigned and they are only assigned to opened facilities. Finally Constraint (5) and Constraint (6) ensures that the only integral values of x_{ij}'s and y_i's are 0 or 1, while Constraint (7) essentially removes any (i, j) pair from consideration if the distance between them is larger than T.

Note that we can now define the LP for MLkSC as

$$\text{Minimize } T \text{ such that SC-LP}(T, k) \text{ is feasible,} \qquad \text{(MLkSC-LP)}$$

where one can find the minimum such T using the standard binary search technique. Similarly, the LP for MSSC can be stated as

$$\text{Minimize } k \text{ such that SC-LP}(T, k) \text{ is feasible.} \qquad \text{(MSSC-LP)}$$

Recall that k is part of the MLkSC input and T is a part of the MSSC input.

For an arbitrary (not necessarily feasible) solution (x, y) to SC-LP(T, k), for $i \in F$ let $L(i, x)$ denote the *fractional load* of facility i with respect to the assignment x, I.e., $L(i, x) := \sum_{j \in C} d(i, j)x_{ij}$. A solution (x, y) to SC-LP(T, k) is called (α, β)-*approximate*, if for every $i \in F$, $L(i, x) \le \alpha T y_i$, and $\sum_{i \in F} y_i \le \beta k$. The proofs of Theorem 3 and Theorem 4 immediately follow from the two theorems on rounding feasible solutions of SC-LP presented below:

▶ **Theorem 5.** *There exists a polynomial time rounding algorithm that, given a feasible solution (x^*, y^*) to* SC-LP(T, k) *and any $\varepsilon \in (0, 1)$, outputs an integral $(O(1/\varepsilon^2), 1 + O(\varepsilon))$-approximate solution to* SC-LP(T, k).

Proof of Theorem 3. Let $\varepsilon \in (0, 1)$ be given. Using standard binary search approach, we can guess the value T^*, such that $\mathrm{OPT}_{\mathrm{MLk}} \le T^* \le 2\mathrm{OPT}_{\mathrm{MLk}}$, by solving MLkSC-LP multiple times for different values of T^* and either finding a feasible fractional solution of load at most T^*, or determining that no such solution exists. Let (x^*, y^*) be the corresponding fractional solution to MLkSC-LP. Observe that (x^*, y^*) is a feasible solution to SC-LP(T^*, k). By Theorem 5, we can round (x^*, y^*) to an integral solution $(\dot{x}, \dot{y})$, which opens at most $(1 + O(\varepsilon))k$ facilities and achieves maximum load at most $O(1/\varepsilon^2)T^*$, and it will take polynomial time. Therefore, $(\dot{x}, \dot{y})$ will be an integral solution to MLkSC-LP with opening at most $(1 + O(\varepsilon))k$ and maximum load at most $O(1/\varepsilon^2)\mathrm{OPT}_{\mathrm{MLk}}$. ◀

▶ **Theorem 6.** *There exists a polynomial time rounding algorithm that, given a feasible solution* (x^*, y^*) *to* SC-LP(T, k) *and any* $\varepsilon \in (0,1)$*, outputs an integral* $(2 + O(\varepsilon), O(1/\varepsilon^2))$*-approximate solution to* SC-LP(T, k)*.*

The proof of Theorem 4 using Theorem 6 is just the same as the proof of Theorem 3 using Theorem 5, omitting the binary search part (as we optimize over k instead of T).

Note that MLkSC-LP closely resembles the LP used in minimum make-span rounding by Lenstra et al. [7]. In fact, for the case where we do not have a restriction on number of opened facilities, we can assume $y_i = 1$ for all $i \in F$, and the LP reduces to the minimum make-span problem, yielding a 2-approximation algorithm. The main difficulty here is to figure out which facilities to open. Once we have an integral opening of facilities, we can still use minimum make-span rounding at a loss of only a factor 2 in the guarantee for minimum load. Thus, our algorithm for MLkSC essentially transforms the initial solution for MLkSC-LP via a series of steps to a solution with integral openings, I.e., $y_i \in \{0,1\}$ for all $i \in F$, and fractional assignments, without violating Constraint 1 by too much.

When we fully open (I.e., set $y_i = 1$) some facilities in the solution, inevitably, we have to close down (set $y_i = 0$) some other partially opened facilities, which requires redistributing their assigned clients' demand to the opened ones. This process is called *rerouting* and is a well-known technique in rounding facility-location-like problems. However, instead of bounding the total load of all facilities, our problem requires bounding each $L(i, x)$ for $i \in F$ separately, and consequently, many facility-location rounding algorithms which use rerouting fail to produce a good solution.

Let x° be the solution we obtain from x after rerouting facility i to facility h. Using triangle inequality $d(h, j) \le d(h, i) + d(i, j)$ for $j \in C$, we bound $L(h, x^\circ)$, the new load of h:

$$L(h, x^\circ) \le L(h, x) + L(i, x) + d(h, i) \sum_{j \in C} x_{ij}.$$

If both $L(h, x)$ and $L(i, x)$ were initially $O(T)$, the new load of h will also be $O(T)$ if and only if the value $d(h, i) \sum_{j \in C} x_{ij} \le d(h, i)|N(i)|$ is also at most $O(T)$ (here $N(i)$ is the set of all clients partially served by i). However, if $d(h, i)|N(i)|$ is large for all other facilities h, a good alternative to rerouting is to open i integrally and assign every client in $N(i)$ to i. We call such facilities *heavy* facilities. There is still an issue if the integral load $\sum_{j \in N(i)} d(i, j)$ is too large compared to T, but we show that we can prevent having too large integral loads in heavy facilities by preceding the rerouting step with additional filtering and preprocessing steps. The filtering step blows-up the load constraint by a $(1 + \varepsilon)$ factor while ensuring that no client is fractionally assigned to far away facilities. The preprocessing step uses techniques similar to those of minimum make-span rounding by Lenstra et al. [7] to ensure that any non-zero fractional assignment x_{ij} to a facility i is at least a constant factor times its opening y_i, while slightly relaxing other constraints.

Once we identify the heavy facilities, we cluster the remaining, non-heavy facilities, and choose which ones should be opened based on the clustering. Then we redistribute the assignments of the remaining facilities to those that were opened. Using the properties of the preprocessed solution and the clustering, and using the fact that none of the non-opened facilities are heavy, we show that the resulting fractional assignment satisfies the constraints up to an $O(1/\varepsilon^2)$ factor violation of load constraints. Hence, the algorithmic result of Theorem 5 follows from the minimum make-span rounding of Lenstra et al. [7], which gives us an integral assignment with maximum load increased at most by another factor of 2

The algorithm for MSSC problem, on a high level, resembles that for MLkSC: We first alter the solution of MSSC-LP to have integral y_i's and fractional x_{ij}'s, allowing the total opening $\sum_{i \in F} y_i$ to be at most $O(1/\varepsilon^2)$ factor larger than the value of MSSC-LP, and then use minimum make-span rounding of Lenstra et al. [7] to obtain the final solution. However, since make-span rounding guarantees only a factor 2 violation in the load constraint, we need to make sure that our modified solution with integral openings and fractional assignments introduces only small error in load constraints. Namely, to ensure that the final solution satisfies $(2+O(\varepsilon))T$ maximum load, before applying the minimum make-span rounding, all the loads must be at most $(1+O(\varepsilon))T$. We ensure this by re-arranging the steps of the algorithm for MLkSC and carefully choosing parameters.

3 $(O(1/\varepsilon^2), 1+O(\varepsilon))$-approximation to $\mathrm{SC}(T,k)$

In this section, we show how to convert a (feasible) fractional solution (x,y) of SC-LP in to a $(O(1/\varepsilon^2), 1+O(\varepsilon))$-approximate solution with integral y values. This together with minimum make-span rounding scheme by Lenstra et al. [7] proves Theorem 5.

3.1 Preprocessing and filtering

Suppose that for each $(i,j)\in F\times C$ we either have $x_{ij}=0$ or $x_{ij}\geq \gamma y_i$ for constant $\gamma\in(0,1)$. Then, if $L(i,x)=\sum_{j\in C} d(i,j)x_{ij}\leq \nu T y_i$ for some constant $\nu\geq 1$, we have $\sum_{j\in N(i)} d(i,j)\leq \frac{\nu}{\gamma}T$. Therefore, if we open i integrally and assign all $N(i)$ to i, the resulting load of i will be $O(T)$. Even though we cannot guarantee the property above for every solution (x,y) to SC-LP(T,k), we can modify (x,y) so that all non-zero assignments x_{ij} satisfy $x_{ij}\geq\gamma y_i$ for some constant $\gamma\in(0,1)$ at the expense of slightly relaxing other constraints of SC-LP. This is exactly the statement of the preprocessing theorem.

▶ **Theorem 7** (Preprocessing). *Let (x,y) be such that, for all $i\in F$, $L(i,x)\leq \mu T y_i$ for some constant $\mu\geq 1$ and all other constraints of* SC-LP(T,k) *on variables x are satisfied. There exists a polynomial time algorithm that, given such solution (x,y) and a constant $\gamma\in(0,1)$, finds a solution (x',y') such that*
1. *$y'=y$, and if $x_{ij}=0$, then $x'_{ij}=0$;*
2. *for every $(i,j)\in F\times C$, $y'_i\geq x'_{ij}$, and if $x'_{ij}>0$, then $x'_{ij}\geq\gamma y'_i$;*
3. *for every $j\in C$, $1\geq\sum_{i\in F} x'_{ij}\geq 1-\gamma$;*
4. *for every $i\in F$, $L(i,x')\leq(\mu+2-\gamma)Ty'_i$.*

That is to say, we can guarantee the property $\{x_{ij}>0 \iff x_{ij}\geq\gamma y_i\}$ by loosing at most γ portion of each client's demand and slightly increasing each facility's load. Loosing a factor of γ demand is affordable for our purposes, as one can meet the demand constraint by scaling each x_{ij} by a factor of at most $1/(1-\gamma)$. Since γ is a constant, this would blow up the load constraint only by an additional constant factor. The proof of Theorem 7 is rather technical and is given in Appendix A.

We now present our rounding algorithm step by step. Let (x,y) be a feasible fractional solution to SC-LP(T,k) and let $\varepsilon\in(0,1)$. By $(\dot{x},\dot{y})$ we denote the final rounded solution with integral $\dot{y}$ and fractional $\dot{x}$.

▶ **Definition 8.** *For $j\in C$, let $D(j):=\sum_{i\in F} d(i,j)x_{ij}$, the average facility distance to j.*

Let $\rho:=\frac{1+\varepsilon}{\varepsilon}$. By applying the well-known filtering technique of Lin and Vitter [9] to (x,y), we construct a new solution $(\hat{x},\hat{y})$, such that $\sum_{i\in F}\hat{y}_i\leq(1+\varepsilon)k$, $L(i,\hat{x})\leq(1+\varepsilon)T\hat{y}_i$ for all $i\in F$, and for every i,j, $\hat{x}_{ij}\leq\hat{y}_i$ and if $\hat{x}_{ij}>0$, then $d(i,j)\leq\rho D(j)$. Applying Theorem 7 to $(\hat{x},\hat{y})$, we obtain solution (x',y') such that
1. $\sum_{i\in F} y'_i\leq(1+\varepsilon)k$,
2. for all (i,j), $y'_i\geq x'_{ij}$, and if $x'_{ij}>0$, then $x'_{ij}\geq\gamma y'_i$ and $d(i,j)\leq\rho D(j)$,

3. for every $j \in C$, $1 \geq \sum_{i \in F} x'_{ij} \geq 1 - \gamma$, and
4. for every $i \in F$, $L(i, x') \leq (\mu + 2 - \gamma) T y'_i = \nu T y'_i$.

Here $\nu := (\mu + 2 - \gamma)$ is a new load bound. We choose $\mu := 1 + \varepsilon$ and $\gamma := \varepsilon/(1+\varepsilon)$, but will keep the parameters unsubstituted, for convenience. It is easy to see from the bounds above that for every $j \in C$, $\sum_{i \in F : x'_{ij} > 0} y'_i \geq 1/(1+\varepsilon)$.

3.2 Opening heavy facilities

We now give an algorithm to choose heavy facilities based on (x', y').

▶ **Definition 9.** *For subsets $F' \subseteq F$, $C' \subseteq C$, for $i \in F'$ denote $N'(i) := \{j \in C' : x'_{ij} > 0\}$, and for $j \in C'$ denote $N'(j) := \{i \in F' : x'_{ij} > 0\}$.*

The algorithm internally maintains two subsets $F' \subseteq F$ and $C' \subseteq C$. Notice that N' changes as the algorithm modifies F' and C'.

▶ **Definition 10.** *A facility $i \in F'$ is* λ-heavy *for $\lambda > 0$, if $\sum_{j \in N'(i)} D(j) > \lambda T$.*

Algorithm 1 opens all λ-heavy facilities for the given value of λ. It starts with $F' = F$ and $C' = C$ and scans F' for λ-heavy facilities. It fully opens every λ-heavy facility $i \in F'$ and assigns all $N'(i)$ integrally to i. Then, it discards i from F' and $N'(i)$ from C', and continues until all facilities are processed.

Algorithm 1 Opening Heavy Facilities.

Input: A solution (x', y'), $\lambda > 0$.
Output: Partial solution $(\dot{x}, \dot{y})$, sets F', C', such that $\sum_{j \in N'(i)} D(j) \leq \lambda T$, $\forall i \in F'$.

Initialize $F' \leftarrow F$, $C' \leftarrow C$
for $i \in F'$ **do**
    if $\sum_{j \in N'(i)} D(j) > \lambda T$ **then**
        Initialize $C(i) \leftarrow N'(i)$
        $F' \leftarrow F' \setminus \{i\}$, $\dot{y}_i \leftarrow 1$
        for $j \in C(i)$ **do**
            $C' \leftarrow C' \setminus \{j\}$, $\dot{x}_{ij} \leftarrow 1$
            for $h \in F \setminus \{i\}$ **do** $\dot{x}_{hj} \leftarrow 0$
return $(\dot{x}, \dot{y})$, F', C'.

Since for each $h \in F'$ we may discard some clients from $N'(h)$ after every step, facilities that were λ-heavy might become non-λ-heavy under updated F' and C'. Lemma 11 shows that this procedure does not open too many facilities and that the load of opened facilities does not exceed T by too much.

▶ **Lemma 11.** *Let F', C' be the sets returned by Algorithm 1. Then $|F \setminus F'| \leq k/\lambda$, and for each facility $i \in F \setminus F'$, $L(i, \dot{x}) \leq \frac{\nu}{\gamma} T$.*

Proof. The set $F \setminus F'$ is exactly the set of facilities integrally opened during Algorithm 1. For $i \in F \setminus F'$, set $C(i)$ in Algorithm 1 is exactly the set of clients, integrally assigned to i by the algorithm. Observe that for every $i, h \in F \setminus F'$, $i \neq h$, the sets $C(i)$ and $C(h)$ are *disjoint*. Hence, by feasibility of (x, y),

$$|F \setminus F'| \cdot \lambda T < \sum_{i \in F \setminus F'} \sum_{j \in C(i)} D(j) \leq \sum_{j \in C} D(j) = \sum_{j \in C} \sum_{i \in F} d(i,j) x_{ij} \leq \sum_{i \in F} T y_i \leq Tk$$

and $|F \setminus F'| < \frac{Tk}{\lambda T} = \frac{k}{\lambda}$. Next, by the properties of solution (x', y'), $\nu T y'_i \geq \sum_{j \in C(i)} d(i,j) x'_{ij} \geq \gamma y'_i \sum_{j \in C(i)} d(i,j)$, implying $L(i, \dot{x}) = \sum_{j \in C(i)} d(i,j) \leq \frac{\nu}{\gamma} T$. ◀

We apply Algorithm 1 with $\lambda := 1/\varepsilon$, and by Lemma 11 this opens at most εk additional facilities. The load of each opened facility is at most $\frac{\nu}{\gamma}T = \frac{(1+\varepsilon)(\mu+2-\gamma)}{\varepsilon}T = O(T/\varepsilon)$. For the returned sets F' and C', $\sum_{j\in N'(i)} D(j) \le T/\varepsilon$ for all $i \in F'$. Moreover, since $j \in C'$ if and only if j was not served by any λ-heavy facility (which got opened), for all $j \in C'$ we have $\sum_{i\in N'(j)} y'_i = \sum_{i\in F: x'_{ij}>0} y'_i \ge 1/(1+\varepsilon)$. Facilities in $F \setminus F'$ are all integral, and it remains to find the integral opening among facilities in F'.

As discussed earlier, if we reroute $i \in F'$ to $h \in F'$, to guarantee a good approximation we have to bound the term $d(h,i)|N'(i)|$. Observe that $\sum_{j\in N'(i)} D(j)$ is an upper bound for $|N'(i)| \min_{j\in N'(i)} D(j)$. Therefore, to get a good bound, we need to choose h for i so that $d(h,i)$ is at most some constant times $\min_{j\in N'(i)} D(j)$. This requires some sophisticated clustering technique and a wise choice of facility h for every such i.

3.3 Clustering

To create an integral opening over F', we partition F' into disjoint clusters, open some facilities in every cluster and reroute the closed ones into opened ones within the same cluster. Our goal is to cluster F' so that, if i and h belong to the same cluster and we reroute h to i, $d(h,i) \le O(\min_{j\in N'(i)} D(j))$. Classic clustering approaches for facility-location-like problems do not work, and to achieve this bound we are required to design a novel approach.

Let $\mathcal{C} \subseteq C'$ be the set of cluster centers. For every $j \in \mathcal{C}$, let $F'(j) \subseteq F'$ be the set of facilities belonging to the cluster centered at j, for $i \in F'$ let $\mathcal{C}(i)$ be the center of the cluster i belongs to (I.e., $i \in F'(j) \iff \mathcal{C}(i) = j$). Figure 1 visualizes the clustering procedure and Algorithm 2 gives its pseudocode. Below we explain each step of the algorithm in detail.

Algorithm 2 Clustering.

Input: Solution (x', y'), sets F' and C'.
Output: Centers $\mathcal{C} \subseteq C'$ and disjoint clusters $F'(s)$ for every $s \in \mathcal{C}$, $\cup_{s\in\mathcal{C}} F'(s) = F'$.

```
Initialize 𝒞 ← ∅, sort j ∈ C′ by the values of D(j) in ascending order
for j ∈ C′ do
    if ∀s ∈ 𝒞 : d(s, j) > 2ρD(j) then
        𝒞 ← 𝒞 ∪ {j}
For all s ∈ 𝒞, initialize F′(s) ← ∅
for i ∈ F′ do
    if ∃s ∈ 𝒞 : i ∈ N′(s) then
        𝒞(i) ← s, F′(s) ← F′(s) ∪ {i}
    else
        Let j := argmin_{r∈N′(i)} D(r)
        Take s ∈ 𝒞, such that D(s) ≤ D(j), d(s, j) ≤ 2ρD(j)
        𝒞(i) ← s, F′(s) ← F′(s) ∪ {i}
return 𝒞, F′(s) for s ∈ 𝒞.
```

The clustering procedure works as follows. First, we form cluster centers $\mathcal{C}$ by scanning $j \in C'$ in ascending order of $D(j)$ and adding j to $\mathcal{C}$ only if there are no other centers in $\mathcal{C}$ within the distance $2\rho D(j)$ from j. Having determined $\mathcal{C}$, we add facilities from F' to different clusters. Most classical clustering approaches would put i into $F'(s)$, if s is closest to i among $\mathcal{C}$. Our approach is different: if $i \in F'$ is serving some $s \in \mathcal{C}$, we add $i \in F'(s)$ regardless the distance $d(i,s)$. Otherwise, we consider $j \in N'(i)$ with *minimum* $D(j)$ (j is not a cluster center), take $s \in \mathcal{C}$ that prevented j from becoming a center, and add i to $F'(s)$. One can easily check that, after Algorithm 2 finishes, for any $s, v \in \mathcal{C}$,

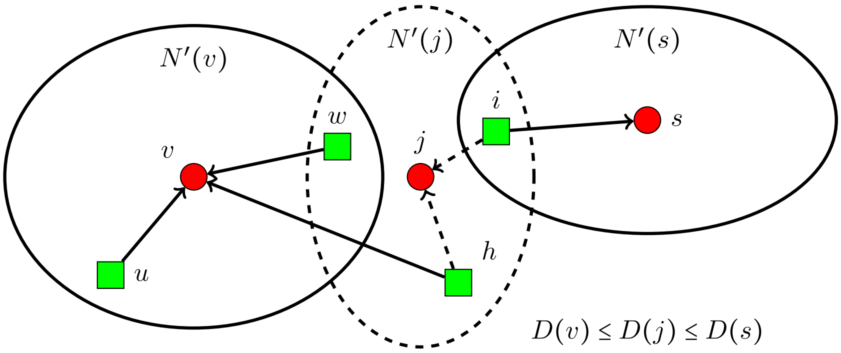

Figure 1 Here, $v, j, s \in C'$, $v, s \in \mathcal{C}$, $j \notin \mathcal{C}$, $u, w, h, i \in F'$. The bold arrow $\rightarrow$ shows that a facility belongs to the cluster centered at that client, the dashed arrow $\dashrightarrow$ shows that a particular client has minimum average distance among all clients served by a facility.

$s \neq v$, $d(s,v) > 2\rho \max\big(D(s), D(v)\big)$, and as a result $N'(s)$ and $N'(v)$, as well as $F'(s)$ and $F'(v)$ are disjoint. Also, for every $j \notin \mathcal{C}$ there exists $s \in \mathcal{C}$ such that $D(s) \le D(j)$ and $d(s,j) \le 2\rho D(j)$, simply by construction of the algorithm. Algorithm 2 allows us to obtain an upper bound on the distance between a facility an its cluster center, represented in terms of minimum average distance of the client served by this facility.

▶ **Lemma 12.** *Let $i \in F'$, let $j = \operatorname{argmin}_{r \in N'(i)} D(r)$. Then $d(i, \mathcal{C}(i)) \le 3\rho D(j)$.*

Proof. Let $\mathcal{C}(i) = s$. There are two cases to distinguish.

- $i \notin N'(s)$ (this case is shown by clients j, v and facility h in Figure 1). By construction of Algorithm 2, client s is exactly the one that prevented j from becoming a cluster center, therefore $D(s) \le D(j)$ and $d(j,s) \le 2\rho D(j)$. Thus, by triangle inequality $d(i,s) \le d(i,j) + d(j,s) \le \rho D(j) + 2\rho D(j) = 3\rho D(j)$.
- $i \in N'(s)$. Then $D(j) = \min_{r \in N'(i)} D(r) \le D(s)$. If $s = j$ or $D(s) = D(j)$, then $d(i,s) \le \rho D(j)$ automatically. Suppose that $s \neq j$, and $D(j) < D(s)$, then $j \notin \mathcal{C}$, as $s \in \mathcal{C}$ and $i \in N'(j) \cap N'(s)$ (this case is shown by clients j, s and facility i in Figure 1). Hence, there exists some $s' \in \mathcal{C}$ that prevented j from becoming a cluster center, so $D(s') \le D(j)$ and $d(s', j) \le 2\rho D(j)$. It is easy to see that $D(j)$ must be strictly greater than zero, and since both s and s' are cluster centers, $d(s', s) > 2\rho D(s)$. So, by triangle inequality,
$$2\rho D(s) < d(s', s) \le d(s', j) + d(i,j) + d(i,s) \le 2\rho D(j) + \rho D(j) + \rho D(s),$$
implying $2\rho D(s) \le 3\rho D(j) + \rho D(s)$ and $D(s) \le 3D(j)$. Since $i \in N'(s)$, it immediately follows that $d(i,s) \le \rho D(s) \le 3\rho D(j)$. ◀

By applying the triangle inequality once more, we get the desired upper bound on the distances between any two facilities within the same cluster.

▶ **Corollary 13.** *Let $i, h \in F'$, such that $\mathcal{C}(i) = \mathcal{C}(h)$. Let $j = \operatorname{argmin}_{r \in N'(i)} D(r)$ and $v = \operatorname{argmin}_{l \in N'(h)} D(l)$. Then $d(i,h) \le 6\rho \max\big(D(j), D(v)\big)$.*

Another useful observation is that $\sum_{i \in F'(s)} y'_i \ge 1/(1+\varepsilon)$ for every cluster center $s \in \mathcal{C}$. It follows from $N'(s) \subseteq F'(s)$ and $\sum_{i \in N'(s)} y'_i \ge 1/(1+\varepsilon)$.

3.4 Rerouting

The last part of our rounding algorithm is opening some facilities in every cluster and rerouting the closed ones. For $s \in \mathcal{C}$ we open $\lfloor (1+\varepsilon) \sum_{u \in F'(s)} y'_u \rfloor$ facilities in cluster $F'(s)$, prioritizing facilities i with minimum values of $\min_{r \in N'(i)} D(r)$. Since $\sum_{u \in F'(s)} y'_u \ge 1/(1+\varepsilon)$, we will open at least one facility in every cluster $F'(s)$ for $s \in \mathcal{C}$. Then, the demand of each closed facility in $F'(s)$ is redistributed at an equal fraction between all the opened ones in $F'(s)$, I.e. we reroute it to all opened facilities in $F'(s)$. This gives us an integral opening $\dot{y}$ over facilities in F' and a fractional assignment $\dot{x}$ over clients in C'.

Lemma 14 shows that by opening $|K_s| = \lfloor (1+\varepsilon) \sum_{u \in F'(s)} y'_u \rfloor$ facilities in cluster $F'(s)$ and rerouting all closed facilities in $F'(s)$, we open at most $(1+3\varepsilon)k$ facilities in total, and the load of every opened facility in F' exceeds T at most by a constant factor.

Algorithm 3 Rerouting.

Input: Solution (x', y'), cluster centers $\mathcal{C}$ and clusters $F'(s)$ for $s \in \mathcal{C}$.
Output: Solution $(\dot{x}, \dot{y})$, sets of opened facilities K_s for $s \in \mathcal{C}$.

for $s \in \mathcal{C}$ **do**
  Initialize $K_s \leftarrow \varnothing$
  Sort $i \in F'(s)$ in ascending order of $\min_{r \in N'(i)} D(r)$
  for $i \in F'(s)$ **do**
    if $y'_i = 0$ **then**
      $\dot{y}_i \leftarrow 0$
    else if $|K_s| + 1 \le \lfloor (1+\varepsilon) \sum_{u \in F'(s)} y'_u \rfloor$ **then**
      $K_s \leftarrow K_s \cup \{i\}$, $\dot{y}_i \leftarrow 1$
      for $j \in N'(i)$ **do** $\dot{x}_{ij} \leftarrow x'_{ij}$
    else
      $\dot{y}_i \leftarrow 0$
      for $r \in N'(i)$ **do**
        $\dot{x}_{ir} \leftarrow 0$
        for $h \in K_s$ **do**
          $\dot{x}_{hr} \leftarrow \dot{x}_{hr} + x'_{ir}/|K_s|$
return $(\dot{x}, \dot{y})$, K_s for $s \in \mathcal{C}$.

▶ **Lemma 14.** *After Algorithm 3, for every facility $h \in F'$, $\dot{y}_h = 1$: $L(h, \dot{x}) \le 3(\nu + 4\rho\lambda)T$. Moreover, $\sum_{h \in F'} \dot{y}_h \le (1+3\varepsilon)k$.*

Proof. Since for every $s \in \mathcal{C}$ we have $\sum_{u \in F'(s)} y'_u \ge 1/(1+\varepsilon)$, $\lfloor (1+\varepsilon) \sum_{u \in F'(s)} y'_u \rfloor \ge 1$ and $|K_s| \ge 1$. After filtering and preprocessing steps, $\sum_{u \in F'} y'_u \le (1+\varepsilon)k$, so

$$\sum_{h \in F'} \dot{y}_h = \sum_{s \in \mathcal{C}} \sum_{h \in K_s} \dot{y}_h \le (1+\varepsilon) \sum_{s \in \mathcal{C}} \sum_{u \in F'(s)} y'_u = (1+\varepsilon) \sum_{u \in F'} y'_u \le (1+\varepsilon)^2 k \le (1+3\varepsilon)k.$$

Next, let $h \in F'$, $\dot{y}_h = 1$, and let $\mathcal{C}(h) = s$. Take $i \in F'(s)$ that was closed by Algorithm 3. The demand of every $r \in N'(i)$ served by i gets split between all opened facilities from K_s at an equal fraction. So, after we reroute i into h, the *additional* load of h is

$$\sum_{r \in N'(i)} d(h,r) \frac{x'_{ir}}{|K_s|} \le \frac{1}{|K_s|} \sum_{r \in N'(i)} d(i,r) x'_{ir} + \frac{d(h,i)}{|K_s|} \sum_{r \in N'(i)} x'_{ir}.$$

Recall that $\sum_{r \in N'(i)} d(i,r) x'_{ir} = L(i, x') \le \nu T y'_i$. Let $v = \mathrm{argmin}_{l \in N'(h)} D(l)$ and $j = \mathrm{argmin}_{r \in N'(i)} D(r)$. Since h was opened, and i was closed, $D(v) \le D(j)$, and by Corollary 13 $d(h,i) \le 6\rho D(j)$. Hence, the *additional* load of h is at most

$$
\begin{aligned}
&\frac{1}{|K_s|}\sum_{r\in N'(i)} d(i,r)x'_{ir} + \frac{d(h,i)}{|K_s|}\sum_{r\in N'(i)} x'_{ir} \le \frac{\nu T y'_i}{|K_s|} + \frac{6\rho D(j)}{|K_s|}\sum_{r\in N'(i)} x'_{ir} \le \\
&\le \frac{\nu T y'_i}{|K_s|} + \frac{6\rho D(j)}{|K_s|}\sum_{r\in N'(i)} y'_i = \frac{y'_i}{|K_s|}\left(\nu T + 6\rho\cdot|N'(i)|D(j)\right) \le \\
&\qquad\qquad \le \frac{y'_i}{|K_s|}\left(\nu T + 6\rho\cdot\lambda T\right) = \frac{y'_i}{|K_s|}(\nu + 6\rho\lambda)T.
\end{aligned}
$$

We used the bound $|N'(i)|\min_{r\in N'(i)} D(r) \le \sum_{r\in N'(i)} D(r) \le \lambda T$ for non-λ-heavy facilities. Hence, the *total additional* load of h, gained after rerouting all closed facilities $i \in F'(s)\smallsetminus K_s$ in its cluster, is at most

$$
\begin{aligned}
&\sum_{i\in F'(s)\smallsetminus K_s}\sum_{r\in N'(i)} d(h,r)\frac{x'_{ir}}{|K_s|} \le \sum_{i\in F'(s)\smallsetminus K_s}\frac{y'_i}{|K_s|}(\nu+6\rho\lambda)T = \\
&= (\nu+6\rho\lambda)T\cdot\frac{\sum_{i\in F'(s)\smallsetminus K_s} y'_i}{\lfloor(1+\varepsilon)\sum_{u\in F'(s)} y'_u\rfloor} \le (\nu+6\rho\lambda)T\cdot\frac{(1+\varepsilon)\sum_{i\in F'(s)} y'_i}{\lfloor(1+\varepsilon)\sum_{u\in F'(s)} y'_u\rfloor} \le \\
&\qquad\qquad \le (2\nu+12\rho\lambda)T.
\end{aligned}
$$

The load of h before rerouting was $L(h,x') \le \nu T y'_h \le \nu T$, so after Algorithm 3 the total load of facility h is $L(h,\dot{x}) \le 3(\nu+4\rho\lambda)T$. This holds for every $h \in K_s$ and every center $s \in \mathcal{C}$. ◀

Now we are ready to complete the analysis of the rounding algorithm.

Proof of Theorem 5. We claim that, having completed all the intermediate steps from filtering and up to Algorithm 3 included, with parameter values $\rho = \frac{1+\varepsilon}{\varepsilon}$, $\gamma = \varepsilon/(1+\varepsilon)$ and $\lambda = 1/\varepsilon$, for the resulting solution $(\dot{x},\dot{y})$ it holds:

1. $\dot{y}$ is integral, and $\sum_{i\in F}\dot{y}_i \le (1+4\varepsilon)k$;
2. for every $j \in C$, $1 \ge \sum_{i\in F}\dot{x}_{ij} \ge 1/(1+\varepsilon)$, and if $j \in C\smallsetminus C'$, $\sum_{i\in F}\dot{x}_{ij} = 1$;
3. for every $i \in F$, $L(i,\dot{x}) \le 12\left(1+\frac{1+\varepsilon}{\varepsilon^2}\right)T\dot{y}_i$.

By Lemma 11, Algorithm 1 could open additional εk facilities, so $\sum_{i\in F\smallsetminus F'}\dot{y}_i \le \varepsilon k$. By Lemma 14, $\sum_{h\in F'}\dot{y}_h \le (1+3\varepsilon)k$. This gives us total opening $\sum_{i\in F}\dot{y}_i \le (1+4\varepsilon)k$.

Next, take $j \in C$. If j was serving some λ-heavy facility i, then $j \in C\smallsetminus C'$, and Algorithm 1 sets $\dot{x}_{ij} = 1$ and $\dot{x}_{hj} = 0$ for all other facilities $h \ne i$. If j did not serve any λ-heavy facility, then $j \in C'$, and we get $1 \ge \sum_{i\in F}\dot{x}_{ij} = \sum_{i\in F} x'_{ij} \ge 1/(1+\varepsilon)$ after rerouting.

Finally, if $i \in F$ was λ-heavy, by Lemma 11 $L(i,\dot{x}) \le \frac{\nu}{\gamma}T \le \frac{4}{\varepsilon}T\dot{y}_i$. Let i be non-λ-heavy, I.e. $i \in F'$. If $\dot{y}_i = 0$, I.e. i is closed, then Algorithm 3 assures that $L(i,\dot{x}) = 0$. If $\dot{y}_i = 1$, then by Lemma 14 we have $L(i,\dot{x}) \le 3(\nu+4\rho\lambda)T \le 3\left(4+4\frac{1+\varepsilon}{\varepsilon^2}\right)T\dot{y}_i = 12\left(1+\frac{1+\varepsilon}{\varepsilon^2}\right)T\dot{y}_i$.

For every $(i,j) \in F'\times C'$, we multiply the assignment variables $\dot{x}_{ij}$ by $1/\left(\sum_{i\in F}\dot{x}_{ij}\right)$. Since $\sum_{i\in F}\dot{x}_{ij} \ge 1/(1+\varepsilon)$, the load of every opened facility in F' gets increased at most by a factor of $1+\varepsilon \le 2$. After this change, $\sum_{i\in F}\dot{x}_{ij} = 1$ and $L(i,\dot{x}) \le 24\left(1+\frac{1+\varepsilon}{\varepsilon^2}\right)T$ for all $j \in C$, $i \in F$.

The solution $(\dot{x},\dot{y})$ has integral opening $\dot{y}$, and every client $j \in C$ is served fully (I.e. $\sum_{i\in F}\dot{x}_{ij} = 1$). By applying minimum makespan rounding algorithm [7], we get an integral assignment with respect to facilities opened in $\dot{y}$, sacrificing another factor of 2 in approximation. We obtain a $\left(48\left(1+\frac{1+\varepsilon}{\varepsilon^2}\right), 1+4\varepsilon\right)$-approximate solution to $\mathrm{SC}(T,k)$ problem, and the whole algorithm clearly runs in polynomial-time. ◀

4 $(2+O(\varepsilon), O(1/\varepsilon^2))$-approximation to $\mathrm{SC}(T,k)$

Similar to the $(O(1/\varepsilon^2), 1+O(\varepsilon))$-approximation to $\mathrm{SC}(T,k)$, our goal is, given some solution (x,y) to SC-LP(T,k), find an integral opening $\dot{y}$ and fractional assignment $\dot{x}$, and then apply minimum makespan rounding [7], which will prove Theorem 6. However, this time we need to assure that $L(i,\dot{x}) \le (1+O(\varepsilon))T$ for every $i \in F$. To achieve this, we use the same steps, applied in different order and with different values of parameters.

4.1 Preprocessing and opening heavy facilities

Let (x,y) be a feasible fractional solution to SC-LP(T,k), and let $\varepsilon \in (0,1)$. Straightahead, we apply preprocessing algorithm from Theorem 7 to (x,y) with parameters $\mu = 1$ and $\gamma = \frac{1}{1+\varepsilon}$. This will give us a solution (x',y') such that

1. $y' = y$, and $\sum_{i\in F} y'_i \le k$,
2. for all $(i,j) \in F\times C$, $y'_i \ge x'_{ij}$ and if $x'_{ij} > 0$ then $x'_{ij} \ge \gamma y'_i = y'_i/(1+\varepsilon)$,
3. for every $j \in C$, $1 \ge \sum_{i\in F} x'_{ij} \ge 1-\gamma = \varepsilon/(1+\varepsilon)$, and
4. for every $i \in F$, $L(i,x') \le (\mu+2-\gamma)Ty'_i = (2+\frac{\varepsilon}{1+\varepsilon})Ty'_i = \nu T y'_i$.

In this algorithm, we overuse the notation and define $D(j)$ with respect to assignment x'.

▶ **Definition 15.** *For $j \in C$, let $D(j) := \sum_{i\in F} d(i,j)x'_{ij}$, the average facility distance to j.*

The definitions of $N'(i)$, $N'(j)$ for $i \in F'$, $j \in C'$, given $F' \subseteq F$ and $C' \subseteq C$, are the same.

▶ **Definition 16.** *For $F' \subseteq F$, $C' \subseteq C$, let $N'(i) := \{j \in C' : x'_{ij} > 0\}$, $N'(j) := \{i \in F' : x'_{ij} > 0\}$.*

▶ **Definition 17.** *A facility $i \in F'$ is* λ-heavy *for $\lambda > 0$, if $\sum_{j\in N'(i)} D(j) > \lambda T$.*

We apply Algorithm 1 to solution (x',y') with $\lambda := \varepsilon^2/15$. Observe that $\sum_{j\in C} D(j) = \sum_{i\in F}\sum_{j\in C} d(i,j)x'_{ij} \le \sum_{i\in F} \nu T y'_i \le \nu T k$. Hence, using a similar analysis as in Lemma 11, we open at most $\frac{\nu}{\lambda}k = O(k/\varepsilon^2)$ additional facilities, and the load of every opened facility is at most $\frac{\nu}{\gamma}T = (1+\varepsilon)\left(2+\frac{\varepsilon}{1+\varepsilon}\right)T = (2+3\varepsilon)T$.

For the returned sets F' and C', $\sum_{j\in N'(i)} D(j) \le \lambda T = \varepsilon^2 T/15$, for every $i \in F'$. As before, it remains to find the integral opening among facilities in F'. However, there may be clients $j \in C'$, for which preprocessing step might have dropped a very huge portion of their demand, as the best bound we have is $\sum_{i\in F'} x'_{ij} \ge \varepsilon/(1+\varepsilon)$. Just for the same reason, the opening $\sum_{i\in N'(j)} y'_i$ may be too small for some clients $j \in C'$, so we cannot apply the clustering and rerouting steps to solution (x',y'), as we did in Section 3, without loosing a lot in both approximation factors, we even do not have any distance upper bounds. We are going to handle these issues by applying a specific filtering step to (x',y'), bounding distances between facilities and clients they serve, as well retrieving the lost demand of every client in C'.

4.2 Filtering

We apply filtering to the restriction of (x',y') on $F'\times C'$, however, the filtering process will be quite different from [9] (the analysis though is similar). We will rely a lot on the fact that we now operate with non-λ-heavy facilities only.

▶ **Definition 18.** *Let $\rho := \frac{(1+\varepsilon)^2}{\varepsilon^2}$. For every $j \in C$ define $F'_j := \{i \in F' : d(i,j) \le \rho D(j)\}$.*

▶ **Lemma 19.** *For every $j \in C'$, $\sum_{i\in F'_j} x'_{ij} \ge 1/(\rho\varepsilon) = \varepsilon/(1+\varepsilon)^2$.*

Proof. Every $j \in C'$ was served by F' only, therefore $D(j) = \sum_{i\in F'} d(i,j)x'_{ij}$. Observe that at most a portion of $1/\rho$ demand of j can be served by facilities not in F'_j. Otherwise,

$$D(j) = \sum_{i\in F'} d(i,j)x'_{ij} \geq \sum_{i\in F'\setminus F'_j} d(i,j)x'_{ij} \geq \rho D(j) \sum_{i\in F'\setminus F'_j} x'_{ij} > \rho D(j)\cdot\frac{1}{\rho} = D(j),$$

a contradiction. Hence, $\sum_{i\in F'\setminus F'_j} x'_{ij} \leq 1/\rho$. Since $\sum_{i\in F'} x'_{ij} \geq \varepsilon/(1+\varepsilon)$ for all $j \in C'$, we have

$$\sum_{i\in F'_j} x'_{ij} = \sum_{i\in F'} x'_{ij} - \sum_{i\in F'\setminus F'_j} x'_{ij} \geq \frac{\varepsilon}{1+\varepsilon} - \frac{\varepsilon^2}{(1+\varepsilon)^2} = \frac{\varepsilon}{(1+\varepsilon)^2} = \frac{1}{\rho\varepsilon}.$$ ◀

We construct a new solution $(\hat{x},\hat{y})$ as follows:

$$\text{for all } (i,j) \in F'\times C', \qquad \hat{x}_{ij} = \begin{cases} 0, & i \notin F'_j; \\ \frac{x'_{ij}}{\sum_{i\in F'_j} x'_{ij}}, & i \in F'_j; \end{cases} \qquad \hat{y}_i = \min\left(1, \rho\varepsilon y'_i\right).$$

Clearly, $\sum_{i\in F'} \hat{y}_i \leq \rho\varepsilon \sum_{i\in F'} y'_i \leq \rho\varepsilon k = O(k/\varepsilon)$. Also, by Lemma 19, $\hat{x}_{ij} \leq \min(1, \rho\varepsilon x'_{ij}) \leq \hat{y}_i$ for every $(i,j) \in F'\times C'$. To bound $L(i,\hat{x})$ for $i \in F'$, recall that i is non-λ-heavy, therefore $\sum_{j\in N'(i)} D(j) \leq \lambda T = \varepsilon^2 T/15$. Since $\hat{x}_{ij} > 0$ if and only if $x'_{ij} > 0$ and $d(i,j) \leq \rho D(j)$,

$$\lambda T\hat{y}_i \geq \sum_{j\in N'(i)} D(j)\hat{y}_i \geq \sum_{\substack{j\in N'(i)\\ s.t.\, \hat{x}_{ij}>0}} D(j)\hat{y}_i \geq \frac{1}{\rho}\sum_{\substack{j\in N'(i)\\ s.t.\, \hat{x}_{ij}>0}} d(i,j)\hat{y}_i \geq \frac{1}{\rho}\sum_{\substack{j\in N'(i)\\ s.t.\, \hat{x}_{ij}>0}} d(i,j)\hat{x}_{ij},$$

implying

$$L(i,\hat{x}) = \sum_{\substack{j\in N'(i)\\ s.t.\, \hat{x}_{ij}>0}} d(i,j)\hat{x}_{ij} \leq \rho\lambda T\hat{y}_i = \frac{\rho\varepsilon^2}{15}T\hat{y}_i = \frac{(1+\varepsilon)^2}{15}T\hat{y}_i.$$

Also, for every $j \in C'$ we now have $\sum_{i\in F'} \hat{x}_{ij} = 1$ and $\sum_{i:\hat{x}_{ij}>0} \hat{y}_i \geq 1$.

Since $\{i \in F' : \hat{x}_{ij} > 0\} \subseteq N'(j)$ and $\{j \in C' : \hat{x}_{ij} > 0\} \subseteq N'(i)$, we will abuse the notation and redefine $N'(i)$ and $N'(j)$ in terms of assignment $\hat{x}$. Let $\hat{\nu} := \frac{(1+\varepsilon)^2}{15}$. It holds for $(\hat{x},\hat{y})$:

1. $\sum_{i\in F'} \hat{y}_i \leq \rho\varepsilon k = O(k/\varepsilon)$,
2. for all $(i,j) \in F'\times C'$, $\hat{y}_i \geq \hat{x}_{ij}$ and if $\hat{x}_{ij} > 0$ then $d(i,j) \leq \rho D(j)$,
3. for every $j \in C'$, $\sum_{i\in F'} \hat{x}_{ij} = 1$ and $\sum_{i\in N'(j)} \hat{y}_i \geq 1$,
4. for every $i \in F'$, $L(i,\hat{x}) \leq \hat{\nu} T\hat{y}_i$.

4.3 Finishing the algorithm

Now we can correctly use our clustering and rerouting algorithms with $(\hat{x},\hat{y})$. We subsequently apply Algorithm 2 and Algorithm 3 to $(\hat{x},\hat{y})$ with newly defined sets N' for F' and C', with corresponding values of of parameters λ, ρ and $\nu \equiv \hat{\nu}$, obtaining the integral opening $\dot{y}$ and possibly fractional assignment $\dot{x}$ over (F',C'). By Lemma 14, for $h \in F'$ with $\dot{y}_h = 1$,

$$L(h,\dot{x}) \leq 3(\hat{\nu} + 4\rho\lambda)T = 3\left(\frac{(1+\varepsilon)^2}{15} + 4\frac{(1+\varepsilon)^2}{\varepsilon^2}\cdot\frac{\varepsilon^2}{15}\right) = (1+\varepsilon)^2 T \leq (1+3\varepsilon)T,$$

and we open at most $(1+\varepsilon)\sum_{i\in F'} \hat{y}_i = O(k/\varepsilon)$ facilities.

Since for every $j \in C$ we have $\sum_{i\in F} \dot{x}_{ij} = 1$, there is no need to modify fractional variables $\dot{x}$ to regain lost demand (as we loose nothing). Observe that all $i \in F\setminus F'$ serve $j \in C\setminus C'$ only, these j are assigned to $i \in F\setminus F'$ integrally, and for all $i \in F\setminus F'$ we have $L(i,\dot{x}) \leq (2+3\varepsilon)T$. Therefore, it remains to obtain integral assignment over (F',C'), where for every $i \in F'$

we have $L(i,\dot{x}) \le (1+3\varepsilon)T$. By applying minimum makespan rounding algorithm [7] to the restriction of $(\dot{x},\dot{y})$ on (F',C'), we get integral assignment, sacrificing a factor of 2 in load approximation for $i \in F'$, resulting in maximum load of the final solution at most $(2+6\varepsilon)T$. Algorithm 1 might have opened at most $O(k/\varepsilon^2)$ additional facilities, so we obtain a $\left(2+6\varepsilon, O(1/\varepsilon^2)\right)$-approximate solution to $\mathrm{SC}(T,k)$ problem, and the whole algorithm clearly runs in polynomial-time, proving Theorem 6.

References

1 Sara Ahmadian, Babak Behsaz, Zachary Friggstad, Amin Jorati, Mohammad R. Salavatipour, and Chaitanya Swamy. Approximation algorithms for minimum-load k-facility location. *ACM Transactions on Algorithms*, 14(2):16:1–16:29, 2018. URL: `http://doi.org/10.1145/3173047`.

2 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. In *Proceedings of the IEEE 58th Annual Symposium on Foundations of Computer Science*, FOCS 2017, pages 61–72, 2017. URL: `http://doi.org/10.1109/FOCS.2017.15`.

3 Esther M. Arkin, Refael Hassin, and Asaf Levin. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, 59(1):1–18, 2006. `doi:10.1016/j.jalgor.2005.01.007`.

4 Jaroslaw Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Transactions on Algorithms*, 13(2):23:1–23:31, 2017. URL: `http://doi.org/10.1145/2981561`.

5 Guy Even, Naveen Garg, Jochen Könemann, R. Ravi, and Amitabh Sinha. Covering graphs using trees and stars. In *Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, APPROX 2003, 2003. `doi:10.1007/978-3-540-45198-3_3`.

6 M. Reza Khani and Mohammad R. Salavatipour. Improved approximation algorithms for the min-max tree cover and bounded tree cover problems. *Algorithmica*, 69:302–314, 2014. `doi:10.1007/s00453-012-9740-5`.

7 Jan Karel Lenstra, David B. Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46(1):259–271, 1990. `doi:10.1007/BF01585745`.

8 Shi Li. A 1.488 approximation algorithm for the uncapacitated facility location problem. In *Proceedings of the 38th International Colloquium on Automata, Languages and Programming*, ICALP 2011, pages 45–58, 2013. `doi:10.1016/j.ic.2012.01.007`.

9 Jyh-Han Lin and Jeffrey Scott Vitter. e-approximations with minimum packing constraint violation. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, STOC 1992, pages 771–782, 1992. `doi:10.1145/129712.129787`.

10 David B. Shmoys and Éva Tardos. Scheduling unrelated machines with costs. In *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 1993, pages 448–454, 1993. URL: `https://dl.acm.org/doi/10.5555/313559.313851`.

A Preprocessing

▶ **Theorem 20** (Theorem 7 restated). *Let (x,y) be such that, for all $i \in F$, $L(i,x) \le \mu T y_i$ for some constant $\mu \ge 1$ and all other constraints of* $\mathrm{SC\text{-}LP}(T,k)$ *on variables x are satisfied. There exists a polynomial time algorithm that, given such solution (x,y) and a constant $\gamma \in (0,1)$, finds a solution (x',y') such that*

1. *$y' = y$, and if $x_{ij} = 0$, then $x'_{ij} = 0$;*
2. *for every $(i,j) \in F \times C$, $y'_i \ge x'_{ij}$, and if $x'_{ij} > 0$, then $x'_{ij} \ge \gamma y'_i$;*
3. *for every $j \in C$, $1 \ge \sum_{i\in F} x'_{ij} \ge 1-\gamma$;*
4. *for every $i \in F$, $L(i,x') \le (\mu + 2 - \gamma) T y'_i$.*

The algorithm we use in Theorem 7 is heavily inspired by the minimum makespan rounding algorithm, introduced by Lenstra et. al in [7]. In a sense, their algorithm achieves the desired property: in minimum makespan problem we have $y_i = 1$ for all $i \in F$, so for $j \in C$ we wish to have either $x'_{ij} = 0$ or $x'_{ij} = 1 = y'_i$. The key difference is that in our case y is *not* integral, which requires several modifications of the original algorithm.

Let (x, y) and $\gamma \in (0,1)$ be given. Let $\tilde{F} \subseteq F$, $\tilde{C} \subseteq C$, let $E \subseteq \tilde{F} \times \tilde{C}$. Consider a bipartite graph $G = (\tilde{F} \cup \tilde{C}, E)$, let $\delta_E(v)$ be the neighbors of $v \in \tilde{F} \cup \tilde{C}$ in G, I.e., for $i \in \tilde{F}$, $\delta_E(i) = \{j \in \tilde{C} : (i,j) \in E\}$, and for $j \in \tilde{C}$, $\delta_E(j) = \{i \in \tilde{F} : (i,j) \in E\}$. For $(i,j) \in \tilde{F} \times \tilde{C}$ we introduce a variable w_{ij}, and numbers $d_j \le 1$ and $L_i \le \mu T y_i$, which can be thought of as the remaining demand of client $j \in \tilde{C}$ and the remaining load of facility $i \in \tilde{F}$ respectively. Given sets $\tilde{F}, \tilde{C}, E$ and numbers d, L, we define the polytope $P(\tilde{F}, \tilde{C}, E, d, L)$ as the solution set of the following feasibility linear program:

$$\begin{aligned} &\sum_{i \in \delta_E(j)} w_{ij} = d_j, && \forall j \in \tilde{C}, \\ &\sum_{j \in \delta_E(i)} d(i,j) w_{ij} \le L_i, && \forall i \in \tilde{F}, \\ &w_{ij} \le \min(y_i, d_j), && \forall (i,j) \in E, \\ &w_{ij} \ge 0, && \forall (i,j) \in E. \end{aligned} \tag{$P(\tilde{F}, \tilde{C}, E, d, L)$}$$

Note that all values y_i for $i \in F$ are fixed, so for every number d_j, $j \in C$, we have either constraint $\{w_{ij} \le y_i\}$ or constraint $\{w_{ij} \le d_j\}$. The extreme points of $P(\tilde{F}, \tilde{C}, E, d, L)$ possess some very important properties, which resemble the properties of the extreme point solutions to the auxiliary program for the minimum makespan rounding algorithm of [7].

▶ **Lemma 21.** *Let w be an extreme point of $P(\tilde{F}, \tilde{C}, E, d, L)$, where $d_j \ge \gamma$ for all $j \in \tilde{C}$. One of the following must hold:*

(a) *there exists $(i,j) \in E$ such that $w_{ij} = 0$,*
(b) *there exists $(i,j) \in E$ such that $w_{ij} = y_i$,*
(c) *there exists $(i,j) \in E$ such that $w_{ij} = d_j$,*
(d) *there eixsts $i \in \tilde{F}$ such that $|\delta_E(i)| \le 1$,*
(e) *there exists $i \in \tilde{F}$ such that $|\delta_E(i)| = 2$ and $\sum_{j \in \delta_E(i)} w_{ij} \ge \gamma y_i$.*

Proof. Suppose that none of (a), (b), (c), or (d) hold. We will show that (e) must hold then.

For all $(i,j) \in E$ we have $0 < w_{ij} < \min(y_i, d_j)$, and for every $i \in \tilde{F}$ we have $|\delta_E(i)| \ge 2$. Since $\sum_{i \in \delta_E(j)} w_{ij} = d_j$ for all $j \in \tilde{C}$, we must also have $|\delta_E(j)| \ge 2$. As w is an extreme point of $P(\tilde{F}, \tilde{C}, E, d, L)$, there exist $\tilde{F}_* \subseteq \tilde{F}$ and $\tilde{C}_* \subseteq \tilde{C}$ such that $\sum_{i \in \delta_E(j)} w_{ij} = d_j$ for all $j \in \tilde{C}_*$, $\sum_{j \in \delta_E(i)} d(i,j) w_{ij} = L_i$ for all $i \in \tilde{F}_*$, $|\tilde{F}_*| + |\tilde{C}_*| = |E|$, and constraints corresponding to $\tilde{F}_*, \tilde{C}_*$ are linearly independent. Since $2|E| = 2|\tilde{F}_*| + 2|\tilde{C}_*| \le \sum_{i \in \tilde{F}_*} |\delta_E(i)| + \sum_{j \in \tilde{C}_*} |\delta_E(j)| \le 2|E|$, for all $i \in \tilde{F}_*$ we must have $|\delta_E(i)| = 2$, as well as $|\delta_E(j)| = 2$ for all $j \in \tilde{C}_*$. Therefore, the subgraph $G[\tilde{F}_* \cup \tilde{C}_*]$ of G induced on $\tilde{F}_* \cup \tilde{C}_*$ is a bipartite union of disjoint cycles.

Let H be a cycle of $G[\tilde{F}_* \cup \tilde{C}_*]$, let $H_{\tilde{F}_*} := H \cap \tilde{F}_*$, $H_{\tilde{C}_*} := H \cap \tilde{C}_*$. Since for all $i \in H_{\tilde{F}_*}$ we have $|\delta_E(i)| = 2$, $\delta_E(i) \subseteq H \cap E$, and similarly, as $|\delta_E(j)| = 2$ for all $j \in H_{\tilde{C}_*}$, $\delta_E(j) \subseteq H \cap E$. Suppose that (e) does not hold, then for all $i \in H_{\tilde{F}_*}$ we have $\sum_{j \in \delta_E(i)} w_{ij} < \gamma y_i$. Consequently,

$$\sum_{i \in H_{\tilde{F}_*}} y_i > \frac{1}{\gamma} \sum_{i \in H_{\tilde{F}_*}} \sum_{j \in \delta_E(i)} w_{ij} = \frac{1}{\gamma} \sum_{(i,j) \in H \cap E} w_{ij} = \frac{1}{\gamma} \sum_{j \in H_{\tilde{C}_*}} \sum_{i \in \delta_E(j)} w_{ij} = \frac{1}{\gamma} \sum_{j \in H_{\tilde{C}_*}} d_j.$$

The last equality follows from $\sum_{i \in \delta_E(j)} w_{ij} = d_j$ for every $j \in \tilde{C}_*$. Since $d_j \ge \gamma$ for all $j \in \tilde{C}$, $d_j \ge \gamma y_i$ for all $(i,j) \in E$. Since H is a cycle in bipartite graph, it has even length, its vertices alternate between $\tilde{F}_*$ and $\tilde{C}_*$, and $|H_{\tilde{F}_*}| = |H_{\tilde{C}_*}|$. Then, we can split vertices of H into disjoint consecutive pairs (i,j), so that $i \in H_{\tilde{F}_*}$, $j \in H_{\tilde{C}_*}$, $(i,j) \in H \cap E$, and apply $d_j \geq \gamma y_i$ for every pair. Therefore, $\sum_{j \in H_{\tilde{C}_*}} d_j \geq \gamma \sum_{i \in H_{\tilde{F}_*}} y_i$, which combined with inequalities above leads to a contradiction. So, there must exist $i \in H_{\tilde{F}_*}$ such that $\sum_{j \in \delta_E(i)} w_{ij} \geq \gamma y_i$, implying (e). ◀

We transform (x,y) into (x',y') using a similar approach as in [7]. On every step $t \geq 1$ of the algorithm, we provide values of parameters $\tilde{F}^t, \tilde{C}^t, E^t, d^t, L^t$ so that polytope $P^t := P(\tilde{F}^t, \tilde{C}^t, E^t, d^t, L^t)$ is nonempty and $d^t_j \geq \gamma$ for $j \in \tilde{C}^t$, and find its extreme point w^t. By Lemma 21, either (a), (b), (c), (d) or (e) cases may occur for w^t. If (a), we set $x'_{ij} \leftarrow 0$, $E^{t+1} \leftarrow E^t \setminus \{(i,j)\}$. If (b), we set $x'_{ij} \leftarrow y_i$, $d^{t+1}_j \leftarrow d^t_j - y_i$, $L^{t+1}_i \leftarrow L^t_i - d(i,j) w^t_{ij}$, $E^{t+1} \leftarrow E^t \setminus \{(i,j)\}$. If (c), we set $x'_{ij} \leftarrow d^t_j$, $d^{t+1}_j \leftarrow 0$, $L^{t+1}_i \leftarrow L^t_i - d(i,j) w^t_{ij}$, $E^{t+1} \leftarrow E^t \setminus \{(i,j)\}$. If (d) or (e), we set $\tilde{F}^{t+1} \leftarrow \tilde{F}^t \setminus \{i\}$. After processing exactly one case (a), (b), (c), (d) or (e), we scan $j \in \tilde{C}^{t+1}$, and if $d^{t+1}_j < \gamma$ for some j, set $\tilde{C}^{t+1} \leftarrow \tilde{C}^{t+1} \setminus \{j\}$, $x'_{ij} \leftarrow 0$ for all (i,j) such that $i \in \delta_{E^{t+1}}(j)$, and then $E^{t+1} \leftarrow E^{t+1} \setminus \{(i,j) : i \in \delta_{E^{t+1}}(j)\}$. If the change of $\tilde{F}^{t+1}, \tilde{C}^{t+1}, E^{t+1}, d^{t+1}$ or L^{t+1} is not mentioned for current case, the values are as in step t, so even though we drop facility i from $\tilde{F}^t$ in case (d) or (e), the edges (i,j) for $j \in \delta_{E^t}(i)$ are still kept in E^{t+1}. Having processed $\tilde{C}^{t+1}$, if $E^{t+1} \neq \varnothing$, we move to step $t+1$ and consider P^{t+1}. Algorithm 4 gives the full pseudocode, summarizing all the steps.

Algorithm 4 Preprocessing.

Input: Initial values of $\tilde{F}$, $\tilde{C}$, E, d, L, parameter $\gamma \in (0,1)$.
Output: An assignment x'.

```
while E ≠ ∅ do
    Find an extreme point w of P(F̃, C̃, E, d, L)
    if ∃(i,j) ∈ E : w_ij = 0 then x'_ij ← 0, E ← E ∖ {(i,j)}
    else if ∃(i,j) ∈ E : w_ij = y_i then
        x'_ij ← y_i, d_j ← d_j − y_i, L_i ← L_i − d(i,j)w_ij, E ← E ∖ {(i,j)}
    else if ∃(i,j) ∈ E : w_ij = d_j then
        x'_ij ← d_j, d_j ← 0, L_i ← L_i − d(i,j)w_ij, E ← E ∖ {(i,j)}
    else if ∃i ∈ F̃ : |δ_E(i)| ≤ 1 then F̃ ← F̃ ∖ {i}
    else if ∃i ∈ F̃ : |δ_E(i)| = 2 and Σ_{j∈δ_E(i)} w_ij ≥ γ y_i then F̃ ← F̃ ∖ {i}
    for j ∈ C̃ do
        if d_j < γ then
            C̃ ← C̃ ∖ {j}, for i ∈ δ_E(j) do x'_ij ← 0, E ← E ∖ {(i,j)}
return x', extended to F × C by adding zero entries
```

It is easy to see that if P^t is nonempty and $d^t_j \geq \gamma$ for $j \in \tilde{C}^t$, the very same holds for P^{t+1} in the next step, unless $E^{t+1} = \varnothing$. Indeed, we manually assure that for all j kept in $\tilde{C}^{t+1}$ the condition $d^{t+1}_j \geq \gamma$ must hold, and the restriction of w^t to the set $E^{t+1} \subseteq E^t$ is a feasible solution to P^{t+1}, by construction of the algorithm. Moreover, if we take $\tilde{F}^1 = F$, $\tilde{C}^1 = C$, $E^1 = \{(i,j) \in F \times C : x_{ij} > 0\}$, $d^1_j = 1$ for $j \in C$ and $L^1_i = \mu T y_i$ for $i \in F$, $d^1_j \geq \gamma$ and P^1 is nonempty, since there is a feasible solution $w_{ij} := x_{ij}$ for $(i,j) \in E^1$. We run Algorithm 4 with these initial values of $\tilde{F}$, $\tilde{C}$, E, d and L given as input, obtaining an assignment x'. By setting $y' := y$, we obtain a solution (x', y').

We claim that Algorithm 4 runs in polynomial-time, and solution (x', y') satisfies all requirements of Theorem 7. By Lemma 21, on every step $t \geq 1$ either (a), (b), (c), (d) or (e) must occur for w^t, the extreme point of P^t. Then, either $|E^t|$, $|\tilde{F}^t|$ or $|\tilde{C}^t|$ is reduced at least by 1 after step t. So, since $|E^1| \leq |F||C|$, after at most $2|F||C|$ steps we will have $E^{t+1} = \varnothing$ for some $1 \leq t \leq 2|F||C|$. Each step t takes only polynomial time to perform, thus the total running time is also polynomial.

Since $E^1 = \{(i,j) \in F \times C : x_{ij} > 0\}$, the only positive coordinates of x' can be (i,j) such that $x_{ij} > 0$, as if $x_{ij} = 0 \iff (i,j) \notin E^1$, Algorithm 4 sets $x'_{ij} = 0$ in the very end. The constraint $\{w_{ij} \le \min(y_i, d_j)\}$ of $P(\tilde{F}, \tilde{C}, E, d, L)$ assures that $x'_{ij} \le y'_i$, for all $(i,j) \in F \times C$. If $x'_{ij} > 0$, then either $x'_{ij} = y'_i$ (case (b)) or $x'_{ij} = d^t_j$ for some step $t \ge 1$ (case (c)). Since $y'_i \le 1$ and for all steps $t \ge 1$ we maintain $d^t_j \ge \gamma$ for all $j \in \tilde{C}$, in both cases we have $x'_{ij} \ge \gamma y'_i$.

Next, if after processing cases for w^t during some step $t \ge 1$ we end up with $d^{t+1} < \gamma$, client j gets discarded from $\tilde{C}^{t+1}$. Since $d^1_j = 1$ initially, by the end of step t we must have assigned at least $1 - \gamma$ portion of j's demand before discarding j having $d^{t+1}_j < \gamma$. Then, after Algorithm 4 finishes, for all $j \in C$ we have $1 \ge \sum_{i \in F} x'_{ij} \ge 1 - \gamma$.

Finally, fix $i \in F$. Observe that if $i \in \tilde{F}^t$ in the beginning of step $t \ge 1$, then

$$\sum_{j \in \delta_{E^t}(i)} d(i,j) w^t_{ij} \le L^t_i = L^1_i - \sum_{C \setminus \tilde{C}^t} d(i,j) x'_{ij} \implies \sum_{C \setminus \tilde{C}^t} d(i,j) x'_{ij} + \sum_{j \in \delta_{E^t}(i)} d(i,j) w^t_{ij} \le \mu T y_i,$$

by feasibility of w^t for polytope P^t. Suppose that after step t facility i gets removed from $\tilde{F}^t$, so $i \notin \tilde{F}^{t+1}$. If case (d) occurred and $|\delta_{E^t}(i)| \le 1$, let $j \in \delta_{E^t}(i)$ be a single client served by facility i. After removing i from $\tilde{F}^t$, the constraint $\{\sum_{j \in \delta_E(i)} d(i,j) w_{ij} \le L_i\}$ is not present in P^{t+1} and all future-step polytopes. So, for any step $r \ge t+1$, the load we may get after obtaining w^r and determining the value of x'_{ij} is at most $d(i,j) w^r \le d(i,j) y_i \le T y_i$ (as $d(i,j) \le T$ for all $x_{ij} > 0$). The total load of facility i becomes $L(i, x') \le (\mu + 1) T y_i$.

If case (e) occurred for this facility i, $|\delta_{E^t}(i)| = 2$ and $\sum_{j \in \delta_{E^t}(i)} w^t_{ij} \ge \gamma y_i$. Let j' and j'' be the two clients belonging to $\delta_{E^t}(i)$. Their contribution to facility i's load on step t is exactly $d(i,j') w^t_{ij'} + d(i,j'') w^t_{ij''}$, which is at most L^t_i. After removing i from $\tilde{F}^t$, the constraint $\{\sum_{j \in \delta_E(i)} d(i,j) w_{ij} \le L_i\}$ is not present in P^{t+1} and all future-step polytopes. So, for any step $r \ge t+1$, the load we may get after obtaining w^r and determining the values of both $x'_{ij'}$ and $x'_{ij''}$ is at most $d(i,j') w^r_{ij'} + d(i,j'') w^r_{ij''} \le d(i,j') y_i + d(i,j'') y_i$. Hence, the *additional* load facility i gained since the end of step t is at most

$$\begin{aligned}
&(d(i,j') y_i + d(i,j'') y_i) - (d(i,j') w^t_{ij'} + d(i,j'') w^t_{ij''}) = \\
&= d(i,j')(y_i - w^t_{ij'}) + d(i,j'')(y_i - w^t_{ij''}) \le T(2y_i - (w^t_{ij'} + w^t_{ij''})) \le \\
&\qquad\qquad \le T(2y_i - \gamma y_i) = (2 - \gamma) T y_i.
\end{aligned}$$

Therefore, the total load of facility i becomes $L(i, x') \le (\mu + 2 - \gamma) T y_i$.

As a result, solution (x', y') and the preprocessing algorithm (Algorithm 4) indeed satisfy all the claimed properties of Theorem 7, thus finishing the proof.

B Hard instances

We first present a hard instance for MLkSC problem. Let R, M be integers, $R \ll M$. Let $k = 2R - 1$, $|F| = 2R$, $|C| = (M + R)R$. F and C are partitioned into R disjoint groups, each has exactly 2 facilities and exactly $M + R$ clients. For $i, h \in F$, $d(i,h) = 1$ if i, h are in the same group, otherwise $d(i,h) = R$. In every group, one facility has M collocated clients (call it M-facility), the other has R collocated clients (R-facility). The instance is illustrated in Figure 2.

There is a feasible fractional solution to MLkSC-LP for this instance with $T = 1$. Open every M-facility fully, and there assign all its collocated clients. Next, open every R-facility to $1 - 1/R$, and let it serve $(1 - 1/R)$-fraction of its collocated clients' demand. The remaining $1/R$ fraction of these clients' demand will be served by M-facility of the same group. It is easy to see that the load of every R-facility is 0, the load of every M-facility is $R \cdot 1/R \cdot 1 = 1$, and the opening is exactly $R \cdot (1 + 1 - 1/R) = 2R - 1 = k$.

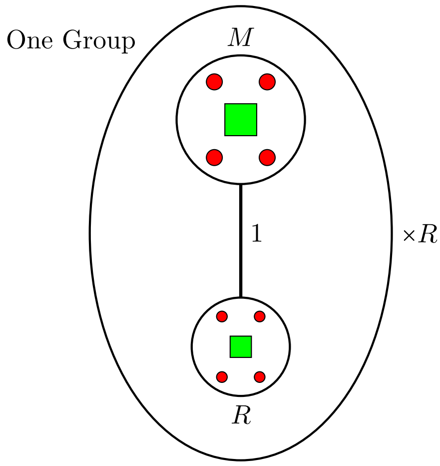

Figure 2 Hard instance for MLkSC-LP.

Consider any integral solution to this instance of MLkSC. If it assigns some client to a facility from different group, maximal load will be at least R. Suppose that all clients are assigned to facilities only from the same group. Since $k = 2R - 1$, there will be at least one group with at most one facility opened, take this group. If M-facility is opened, both its clients and clients of R-facility must be assigned to M-facility fully, resulting in its load $R \cdot 1 \cdot 1 = R$. Similarly, if R-facility is opened, maximum load will be at least $M \gg R$. Hence, the load of any integral star cover of size k is at least R. Furthermore, even if we allow opening $(1+\varepsilon)k$ facilities for $\varepsilon = 1/(2R)$, since

$$(1+\varepsilon)k = \left(1 + \frac{1}{2R}\right)(2R-1) = 2R - \frac{1}{2R} < 2R,$$

there will still be a group with at most one facility opened, resulting in maximum load at least $R = T/(2\varepsilon)$, where $T = 1$ is maximal fractional load. It follows that if T^* is an optimal load to MLkSC-LP, any integral $(1+\varepsilon)k$ star cover of (F, C) has load is at least $\Omega(1/\varepsilon)T^*$.

Now, we move to a hard instance for MSSC. For integer N, let $|F| = N$ and $|C| = N + 1$, the load bound $T \geq 1$ is arbitrary. Both $F = \{i_1, \ldots, i_N\}$ and $C = \{J, j_1, \ldots, j_N\}$ are vertices of a bipartite graph, and the metric d is a shortest-path metric. For every $1 \leq r \leq N$ we have an edge (i_r, j_r) or length $d(i_r, j_r) = (1 - 1/N)T$. Also, every facility i_r for $1 \leq r \leq N$ is connected to a "central" client J by an edge of length $d(i_r, J) = T$. The instance is illustrated in Figure 3.

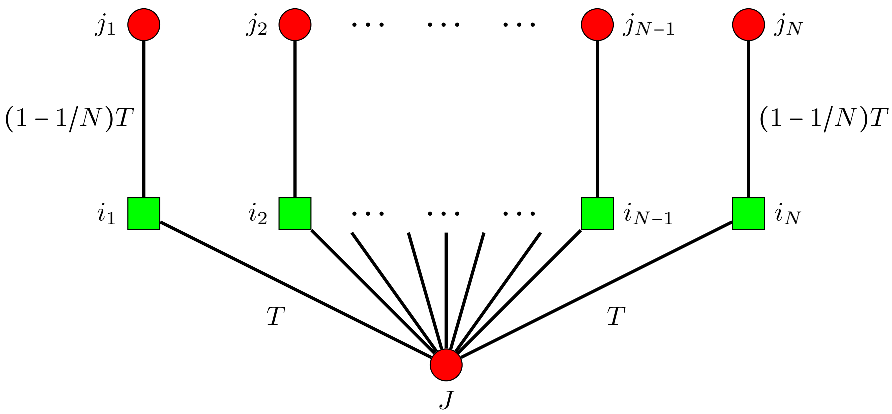

Figure 3 Hard instance for MSSC-LP.

It is easy to see that in any integral solution to MSSC-LP every client j_r for $1 \leq r \leq N$ can be served only by facility i_r. Furthermore, client J should also be served fully, so it should be assigned to one of $i \in F$. Therefore, even if we open all facilities in F fully, for some facility $i \in F$ which gets J assigned to it, the load will be at least $(2 - 1/N)T$. This means that there is no feasible integral solution to MSSC-LP, and any integral solution violates the maximum load constraint at least by a factor of $(2 - 1/N)$.

On the other hand, there exists a feasible *fractional* solution to MLkSC-LP for this instance. We open all i_r for $1 \leq r \leq N$ and assign j_r fully to it. Also, client J gets served by all $i \in F$ at an equal fraction of $1/N$. In this solution, the load of every facility $i \in F$ is exactly T.