S©MR

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ
Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru
ISSN 1813-3304

Vol. 20, No. 2, pp. 144–145 (2023) УДК 519.8
https://doi.org/10.33048/semi.2023.20.??? MSC 90X08

FACILITY LOCATION PROBLEM WITH SOFT UNIFORM CAPACITIES ON PATHS AND TREES

O. Y. TSIDULKO

Communicated by ??

Abstract: We consider the network multiple allocation Soft-Capacitated Facility Location Problem (Soft-CFLP). In this problem, clients and facilities are located in the vertices of a given transportation graph. Each client has an integer demand which is allowed to be jointly served by several facilities. At each facility location one may open a facility of given integer capacity and opening cost. In the soft-capacitated setting multiple copies of a facility are allowed to be opened at the same location. Transportation costs are determined by shortest-path distances between clients and facilities. The goal is to open a set of facilities to serve all client demand, while minimizing total opening and transportation cost.

We show that, in terms of exact solvability on tree-like graphs, the Soft-CFLP is not substantially simpler than the more thoroughly studied CFLP with hard capacities. Namely, being trivially NP-hard even on single-vertex graphs, Soft-CFLP can be solved in pseudopolynomial time on graphs of fixed treewidth. For the case of uniform (identical) capacities, we prove that the problem is NP-hard on trees, but can be solved in polynomial time on paths.

Keywords: Facility Location, Soft capacities, Polynomial-time algorithm, NP-hardness, Treewidth, Tree, Path.

The study was carried out within the framework of the state contract of the Sobolev Institute of Mathematics (project FWNF-2026-0021).

Received May, ??, 2026, Published ??, ??, ????

1 Introduction

Facility location problems (FLPs) are an extensively studied class of problems in operations research, which arise in numerous application domains, including supply chain and logistics network design, emergency service planning, healthcare delivery, telecommunications, transportation systems, and clustering in data mining and machine learning, where cluster centers can be interpreted as facilities serving assigned data points (see, e.g., [12]).

In the classical Uncapacitated FLP we are given a set of n clients each having a positive demand, a set of m facility locations each having an opening cost, and the transportation costs for delivering a unit of demand from each facility to each client. The task is to choose which facilities to open so as to minimize total facility opening and transportation cost. This problem is long known to be strongly NP-hard [9] and in the general (non-metric) case cannot be approximated with a factor better than $O(\log n)$ [8]. In the network FLP clients and possible facility locations are associated with vertices of a given edge-weighted transportation graph $G = (V, E)$. The cost of serving a unit of demand of client at $u \in V$ from a facility at $v \in V$ is defined as the length of a shortest path between u and v in G. Thus, the graph induces a shortest-path metric on the vertex set, and the problem can be regarded as a network-based case of the metric facility location problem. This is closely related to the standard metric formulation, in which service costs are specified between candidate facilities and clients and satisfy the triangle inequality.

The real-world applications commonly impose capacity constraints on facilities, that is, restrictions on the amount of demand each facility can serve. The literature typically distinguishes three Capacitated FLP (CFLP) variants: the hard-capacitated case, the soft-capacitated case, and the more general universal case. In the first two cases, for each location i the facility opening cost f_i and capacity a_i are given constants. In case of hard capacities, at most one facility can be opened at each location i, incurring an opening cost f_i. In case of soft capacities, an arbitrary number of identical facility copies can be opened at each location i, with the opening cost paid for each copy. The so-called universal facility location problem introduced in [13] generalizes both settings by allowing to choose the capacity a_i to allocate to facility i with the opening cost specified as a function $f_i(a_i)$ of the allocated capacity. For each of the settings above one can also consider the multiple allocation (splittable) variant, in which each client can be jointly served by several facilities, and the single allocation (unsplittable) variant, where each client should be served from one facility. The single allocation CFLPs are naturally harder: even when the subset of facilities to be opened is decided, the assignment of clients to facilities contains the NP-hard Bin Packing problem. In what follows, unless stated otherwise, we focus on the multiple-allocation setting.

In the literature, CFLPs have been extensively studied from the perspective of approximation algorithms. For reference, see the recent survey [17], which discusses approximation algorithms for CFLPs with soft and hard capacities, both single and multiple allocation statements, as well as related variants like prize-collecting, robust, and k-level facility location problems. With respect to approximation algorithms, the Soft-CFLP seems to be easier than CFLP with hard capacities: Soft-CFLP with unit client demands in general graphs admits a 2-approximation algorithm [14], while for a similar hard-capacitated variant the currently best approximation factors are 5 [6] and 3 in case of uniform capacities [5].

From the perspective of exact algorithms and polynomial solvable cases, the most studied is the Uncapacitated FLP. It is known to be solved in $O(n)$ time on path graphs [15], in $O(n \log n)$ time on trees [16], in $O(n^{\mathrm{tw}+1})$ time on graphs of fixed treewidth tw [1, 11], where n is the number of vertices of the graph. In contrast, capacitated variants are significantly less tractable. The CFLP with hard capacities is NP-hard even on graphs with a single vertex, provided that several clients and facilities may be located at that vertex, since it contains the NP-hard Knapsack problem as a special case. On the other hand, the CFLP with uniform hard capacities can be solved in $O(m^4 n^2)$ time[1] on paths [2], where n is the number of clients and m is the number of facility locations; however, the problem is already NP-hard on star trees [4]. For soft capacities, it is easy to see that the Soft-CFLP is also NP-hard even on graphs with a single vertex, since it contains the Unbounded Knapsack problem (with unlimited item multiplicities). Nevertheless, to the best of our knowledge, the complexity of the Soft-CFLP with uniform capacities, which we refer to as the Soft-UCFLP, has not been previously analyzed for restricted graph classes.

In this paper, we show that, in terms of exact solvability on tree-like graphs, the Soft-CFLP is not substantially simpler than the CFLP with hard capacities. Our main negative result establishes that the Soft-UCFLP remains NP-hard on trees. On the positive side, by deriving additional structural properties of optimal solutions on paths, we prove that the Soft-UCFLP can be solved in $O(n^3 + n^2 m)$ time on paths, which is lower than the running time $O(m^4 n^2)$ of the known algorithm for the related hard-capacitated case. We also show that the more general Soft-CFLP admits a pseudopolynomial-time algorithm on graphs of bounded treewidth.

2 Preliminaries

2.0.1. Problem Statement. In the network multiple-allocation Soft-CFLP we are given a graph $G = (V, E)$, edge costs $c : E \to \mathbb{N} \cup \{0\}$, set of facility types $I = \{1, \ldots, m\}$ together with a set $\mathcal{F} = \{v_1, \ldots, v_m\} \subseteq V$ of selected vertices where facilities of each type can be located in G, and a set of clients

---

[1] In the Appendix, we identify an error in several subsequent improvements [3, 4] of this algorithm.

$J = \{1, \dots, n\}$ together with a set of vertices $\mathcal{C} = \{u_1, \dots, u_n\} \subseteq V$ where these clients are located. At each $v_i \in \mathcal{F}$ an arbitrary number of facilities of type i can be open, each of an opening cost f_i and capacity a_i. Each client $j \in J$ has a demand b_j. Numerical values $a_i, f_i, b_j \in \mathbb{N} \cup \{0\}$ for all $i \in I, j \in J$. For each $e \in E$ the cost $c(e)$ corresponds to the cost of transportation of a product unit along e, and the cost of transporting a product unit from the facility at v_i to the client at u_j is defined as $d(v_i, u_j) = \sum_{e \in P_{ij}} c(e)$, where P_{ij} is the shortest path between vertices v_i and u_j in G. Each client can be jointly served by several facilities.

The goal is to find the facility opening plan $y = (y_i)$, where y_i is the number of open facilities of type i, and the assignment $x = (x_{ij})$, where x_{ij} is the amount of product delivered to the client j from the location i, that satisfies all client demands, while the total facility opening and transportation cost are minimized:

$$\sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} d(v_i, u_j) \cdot x_{ij} \to \min_{x,y}, \tag{1}$$

subject to

$$\sum_{i=1}^{m} x_{ij} = b_j, \ j \in \{1, \dots, n\}, \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} \le a_i y_i, \ i \in \{1, \dots, m\}, \tag{3}$$

$$x_{ij} \ge 0, y_i \in \mathbb{N} \cup \{0\}, \ i \in \{1, \dots, m\}, j \in \{1, \dots, n\}. \tag{4}$$

Conditions (2) assure that the demand of each client is satisfied, while (3) stands for the total capacity of open facilities of type i is not exceeded. We will denote by (y, x) a feasible solution of the problem, with $y = (y_i)$ and $x = (x_{ij})$. If $y_i \in \{0, 1\}$, we get the hard capacitated CFLP. When $a_i = a$ for all $i \in \mathcal{I}$ we call the it the *Uniform Capacitated FLP*.

2.0.2. Facility Location definitions and properties. We say that a client j is served from location i in a solution of (1)–(4) if $x_{ij} > 0$. Although the formulation (1)–(4) does not introduce separate variables for individual facility copies opened at the same location, we will slightly abuse notation and refer to these copies as distinct facilities. An open facility is called *fully utilized* if its entire capacity is used, and *underutilized* otherwise. Given a feasible solution of (1)–(4), w.l.o.g. we may assume that, at each location i there are $\lfloor \sum_j x_{ij}/a_i \rfloor$ fully utilized open facilities, and at most one underutilized facility if $\sum_j x_{ij} \not\equiv 0 \pmod{a_i}$. Indeed, if two underutilized facilities are open at the same location, their assigned demand can be consolidated into one underutilized copy and probably one fully utilized copy, without increasing the solution cost.

Proposition 1. *Without lost of generality, we can assume that the optimal solution of (1)–(4) no client j is served by two underutilized facilities, since demand can be shifted from the facility with larger distance to the client to the one with smaller distance, until either one of the two assignments becomes zero or one of the facilities becomes fully utilized.*

Proposition 2. *For (1)–(4), if the integer vector (y_i) specifying the number of opened facilities is fixed, then an optimal assignment $x = (x_{ij})$ can be found in polynomial time by solving a transshipment problem. Moreover, with fixed (y_i), the constraint matrix corresponding to (2)–(3) is totally unimodular, so if all capacities a_i and demands d_j are integers, there exists an optimal assignment (x_{ij}) with integer values.*

2.0.3. Graph theory definitions. We consider simple undirected graphs $G = (V, E)$ with a set $V(G) := V$ of *vertices* and a set $E(G) := E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ of *edges*. Degree of a vertex is its number of incident edges. A *simple path* from u to v is a sequence $P = (u = v_0, e_1, v_1, e_2 \dots, e_{k-1}, v_k = v)$ where each $e_i = \{v_i, v_{i+1}\}$ and $v_i \neq v_j$ for $0 \leq i \neq j < k$. A *tree* is a connected acyclic graph. We will also use graphs of bounded treewidth, which generalize trees in a structural sense.

Definition 1. *A* tree decomposition $\mathbb{T} = (T, \beta)$ *of a graph* $G = (V, E)$ *consists of a tree* T *and a function* $\beta \colon V(T) \to 2^V$ *that associates each* node x *of the tree* T *with a subset* $B_x := \beta(x) \subseteq V$, *called a* bag, *such that:*

1. *for each vertex* $v \in V$, *there is a node* x *of* T *with* $v \in B_x$,
2. *for each edge* $\{u, v\} \in E$, *there is a node* x *of* T *with* $\{u, v\} \subseteq B_x$,
3. *for each* $v \in V$ *the nodes* x *with* $v \in B_x$ *induce a subtree of* T.

The width *of* $\mathbb{T}$ *is* $w(\mathbb{T}) := \max_{x \in V(T)} |B_x| - 1$. *The* treewidth *of* G *is* $tw(G) := \min\{w(\mathbb{T}) \mid \mathbb{T}$ *is a tree decomposition of* $G\}$.

For example, if G is a tree, then $\text{tw}(G) = 1$; if G is a cycle, then $\text{tw}(G) = 2$; and if G is a complete n-vertex graph, then $\text{tw}(G) = n - 1$. Graphs of treewidth k are also known as partial k-trees [7]. A *path decomposition* is a tree decomposition whose underlying decomposition tree T is a path. Similarly, the *pathwidth* of a graph is the minimum width of a path decomposition of the graph.

3 Hardness results

We first make two simple hardness observations. The single-allocation Soft-UCFLP, and hence the single-allocation Soft-CFLP, is strongly NP-hard even on a single-vertex graph, since it contains the Bin Packing problem [9]: facility copies correspond to bins, while client demands correspond to item sizes. The multiple-allocation Soft-CFLP with arbitrary capacities is also NP-hard even on a single-vertex graph. Indeed, it contains the minimization version of the Unbounded Knapsack problem [9]: facility types correspond to item types, their opening costs correspond to item weights, their capacities

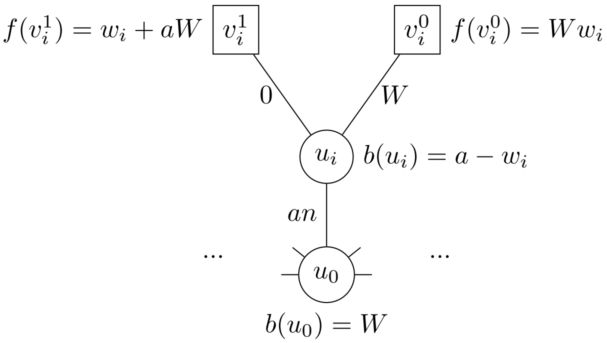

FIG. 1. Hard instance I' of Soft-UCFLP on a tree. Each square denotes a vertex v in which a facility of capacity a can be opened at cost $f(v)$. Each circle denotes a vertex u in which a client of demand $b(u)$ is located. The cost of transporting a unit of product is indicated next to each edge.

correspond to item profits, and the single client demand gives the required profit threshold.

We now turn to the main hardness result of this section and show that, even under uniform capacities, the multiple-allocation Soft-CFLP remains NP-hard on trees.

Theorem 1. *Multiple allocation Soft-UCFLP is NP-hard on trees.*

Proof. To prove the theorem we propose a polynomial-time reduction from the known NP-complete Partitioning problem [9]: given a set of n positive integers $w_1 \geq \cdots \geq w_n > 0$ and an integer budget $W > 0$, the goal is to decide if there is a subset $S \subseteq [n]$ such that

$$\sum_{i \in S} w_i = W. \tag{5}$$

Let $I = (w_1, \ldots, w_n, W)$ be an instance of the Partitioning problem with n input integers. W.l.o.g., we assume that $\sum_{i=1}^n w_i > W > w_1$, since if $\sum_{i=1}^n w_i < W$, the answer is trivially "no"; if $\sum_{i=1}^n w_i = W$, the answer is "yes"; if $w_1 > W$, integer w_1 can be safely discarded from the input. We construct an equivalent instance I' of the decision version of the Soft-UCFLP as follows (see Fig. 1). Let $G = (V, E)$ be the input graph with $V = \{u_0\} \cup \{u_i, v_i^1, v_i^0 \mid i = 1, \ldots, n\}$ and $E = \{\{u_i, u_0\}, \{u_i, v_i^1\}, \{u_i, v_i^0\} \mid i = 1, \ldots, n\}$. Let $a = 2W$ be the capacity of each facility. Let the budget of a desired solution be $F = (2an + 1)W$. Let the edge costs of $\{u_i, u_0\}$, $\{u_i, v_i^0\}$ and $\{u_i, v_i^1\}$ be an, W and 0, respectively. Let the client demand at u_i be $b(u_i) := a - w_i$ for each $i = 1, \ldots, n$, and $b(u_0) := W$. For each $i = 1, \ldots, n$, let the facility opening costs at v_i^0 and v_i^1 be $f(v_i^0) := W \cdot w_i$ and $f(v_i^1) := w_i + W \cdot a$.

Now we show that instance I of the Partitioning problem is a "yes" instance iff there is a feasible solution for the constructed Soft-UCFLP instance I' of total cost at most F.

($\Leftarrow$) If I has a solution S satisfying (5), consider the following feasible solution for I'. For each $i \in S$ open one facility at v_i^1, serve the full demand $b(u_i) = a - w_i$ of a client at u_i from it and send the remaining w_i units of product to the client at u_0. For each $i \in \{1, \dots, n\} \setminus S$ open one facility at v_i^0 and only serve the full demand of a client at u_i from it. Note that the client demand at u_0 is fully satisfied, since (5) holds. The total cost of this solution is:

$$\sum_{i \in S} \big(w_i + aW + anw_i\big) + \sum_{i \notin S} \big(w_i W + W(a - w_i)\big) = (an+1) \sum_{i \in S} w_i + anW = F,$$

where the last equality is due to (5).

($\Rightarrow$) Assume that I' has a feasible solution of total cost at most F. Then we show that there exists a minimum-cost solution for I', in which

(i) for each $i = 1, \dots, n$, there is at least one facility open at $\{v_i^1, v_i^0\}$;
(ii) for each $i = 1, \dots, n$, full demand of the client at u_i is served from the closest open facility;
(iii) there are at most n open facilities.

To show (i), note that in any feasible solution one have to pay at least anW to serve the client at u_0. Now, if for some $i = 1, \dots, n$ no facilities at v_i^1 and v_i^0 are open, the transportation costs for serving the client at u_i will be at least $2an(a - w_i)$, and such solution will not fit the budget F, since:

$$anW + 2an(a - w_i) - F = anW + 2an(a - w_i) - (2an + 1)W =$$
$$2an(a - w_i) - anW - W > 4nW^2 - 2nW^2 - W > W^2 n > 0,$$

where $a = 2W$, $w_i < W$ for all $i \in [n]$ and, thus, $a - w_i > W$.

To show (ii), suppose the opposite: in any least cost solution there is a client u_i for some $i = 1, \dots, n$, which receives only $x_i \in [0, a - w_i)$ demand units from the closest open facility at vertex v, and $v \in \{v_i^1, v_i^0\}$ by (i). Consider such solution with the maximal value of x_i. Since $x_i < a - w_i$, there should be another open facility serving u_i in this solution, let it be located at vertex v' and be sending $x_i' > 0$ demand units to u_i. Now, if capacity of the facility at v in this solution is not fully utilized, there is $r > 0$ such that sending $x_i + r$ demand units from v and $x_i' - r$ demand units from v' to u_i is still feasible and does not increase the solution cost, since $d(v', u_i) \geq d(v, u_i)$, leading to the contradiction with maximality of x_i. If capacity a of the facility at v is fully utilized, there has to be another client at some vertex $u \neq u_i$, which receives $x_u > 0$ demand units from v and some $x_u' \geq 0$ demand units from v'. Let $r = \min\{x_i', x_u\} > 0$. Then it is feasible to modify the current solution such that the client at u_i receives $x_i + r$ and $x_i' - r$ demand units from v and v', respectively, while the client at u receives $x_u - r$ and $x_u' + r$ demand units from v and v', respectively. The additional cost brought by this modification is

$$r\big(d(v,u_i) - d(v',u_i) + d(v',u) - d(v,u)\big) \le 0,$$

since

$$d(v',u)-d(v,u) = \begin{cases} d(v',u_i) - d(v,u_i), & \text{if } v' \in \{v_i^1, v_i^0\}, \\ d(v',u_i) - d(v,u_i) - 4an, & \text{if } u = u_k, k \ne i, v' \in \{v_k^1, v_k^0\}, \\ d(v',u_i) - d(v,u_i) - 2an, & \text{otherwise,} \end{cases}$$

which leads to the contradiction with maximality of x_i.

To show (iii), consider set S of n open facilities that serve the full demand of clients at $u_1, \dots, u_n$ in the least cost solution satisfying (i)-(ii). The total capacity of S is an, and after serving all clients at $u_1, \dots, u_n$ the remaining available total amount of resources is $\sum_{i=1}^n w_i > W$, which is enough to serve the client at u_0. We now show that it is suboptimal to open any extra facilities to serve u_0. Let C be the smallest possible transportation cost of serving u_0 from S after all u_i were served. Consider the case where one opens an additional facility at a vertex v to serve $\alpha \in (0, a]$ demand of u_0 from it instead of S. The total transportation costs for serving u_0 together with the additional cost for opening a facility at v will be at least

$$\begin{aligned} (C - \alpha(an+W)) + \alpha(an+W) + Ww_n > C, \quad & \text{if } v \in \{v_i^0 \mid i = 1, \dots, n\}, \\ (C - \alpha(an+W)) + \alpha an + w_n + aW > C, \quad & \text{if } v \in \{v_i^1 \mid i = 1, \dots, n\}. \end{aligned}$$

Finally, consider a feasible solution satisfying (i)–(iii), and let $S \subseteq \{1, \dots, n\}$ be such that in this solution facilities are open at v_i^1 for $i \in S$ and at v_i^0 for $i \in \{1, \dots, n\} \setminus S$. According to Proposition 2, there is an integer demand assignment for our choice of open facilities. Thus, let $\alpha_i \in \{0, 1, \dots, w_i\}$ be an integer amount of product sent from i-th open facility to u_0, where $\sum_{i=1}^n \alpha_i = W$. Since the total solution cost of a feasible solution is at most F, it holds that:

$$\sum_{i\in S}\Big(w_i + aW + an\alpha_i\Big) + \sum_{i\in\{1,\dots,n\}\setminus S}\Big(Ww_i + W(a-w_i) + (W+an)\alpha_i\Big) =$$

$$\sum_{i\in S} w_i + \sum_{i\in\{1,\dots,n\}\setminus S} W\alpha_i + an\sum_{i=1}^n \alpha_i + anW = \sum_{i\in S} w_i + \sum_{i\in\{1,\dots,n\}\setminus S} W\alpha_i +$$

$$2anW \le F = (2an+1)W,$$

and thus,

$$\sum_{i\in S} w_i + \sum_{i\in\{1,\dots,n\}\setminus S} W\alpha_i \le W. \tag{6}$$

It is easy to see that $\alpha_i \ge 1$ for any $i \in \{1, \dots, n\} \setminus S$ violates (6), therefore, $\alpha_i = 0$ for all $i \in \{1, \dots, n\} \setminus S$. Thus, on the one hand,

$$W = \sum_{i=1}^n \alpha_i = \sum_{i\in S} \alpha_i \le \sum_{i\in S} w_i,$$

and on the other hand, inequality (6) reduces to $\sum_{i\in S} w_i \le W$, whence it follows that S is a feasible solution for the Partitioning problem (5). $\square$

4 Soft-CFLP on graphs of fixed treewidth

In this section we show that Soft-CFLP can be solved in pseudopolynomial time on graphs with constant treewidth even in case of non-uniform capacities. To that end, we give a straightforward pseudopolynomial-time reduction from Soft-CFLP to the so-called Restricted Facility Location Problem (RFLP) studied in [10]. The difference between RFLP and CFLP is that in RFLP the facilities have unbounded capacities, while for each edge e its capacity $q(e) \in \mathbb{N} \cup \{0, \infty\}$ is given, and the total amount of product transported along each edge e in the solution should not exceed $q(e)$. For the multiple allocation RFLP on graphs of constant treewidth paper [10] proposed a pseudopolynomial-time algorithm based on dynamic programming on the tree decomposition.

Proposition 3. [10, Theorem 1] *The multiple allocation RFLP can be solved in $O(N \cdot D^{2tw})$-time, where N is the number of vertices in the input graph G, tw is the treewidth of G and $D = \min\{\sum_{j=1}^{n} b_j,\ \max_{e\in E} q(e)\}$.*

Theorem 2. *Soft-CFLP on graphs of constant treewidth admits a pseudopolynomial-time algorithm.*

Proof. Consider an instance $I = (G, \mathcal{F}, \mathcal{C}, \{f_i\}_{i=1}^{m}, \{a_i\}_{i=1}^{m}, \{b_j\}_{j=1}^{n})$ of the Soft-CFLP. Let $B = \sum_{j=1}^{n} b_j$ be the total client demand. Note that in I it is suboptimal to open more than $m_i = \lceil B/a_i \rceil$ facilities of type i for each $i = 1, \ldots, m$, since their total capacity $a_i \cdot \lceil B/a_i \rceil$ is already enough to serve total client demand from v_i.

Given I, construct an instance $I' = (G', \mathcal{F}', \mathcal{C}, \{f'_i\}_{i=1}^{m'}, \{q(e)\}, \{b_j\}_{j=1}^{n})$ of the RFLP as follows. To obtain graph G', take graph G and for each $v_i \in \mathcal{F}$ add new vertices $v_i^1, \ldots, v_i^{m_i}$ each connected by a zero-cost edge with v_i. For an edge $e \in E(G')$, if $e = \{v_i, v_i^k\}$ set its capacity $q(e) = a_i$, $i = 1, \ldots, m$, $k = 1, \ldots, m_i$, and $q(e) = \infty$, otherwise. $\mathcal{F}' = \bigcup_{i=1}^{m} \bigcup_{k=1}^{m_i} v_i^k$ is a set of vertices in which facilities of unbounded capacities can be open in I', and the cost of opening a facility at v_i^k is f_i, $i = 1, \ldots, m$, $k = 1, \ldots, m_i$.

A facility open at v_i^k at cost f_i in I' can provide at most a_i product to the solution, since it is connected to the rest of G' through a single edge of capacity a_i; this product can then be passed to vertex v_i at zero cost. This is equivalent to opening one copy of i-th type facility with capacity a_i at vertex v_i in I. For each $i = 1, \ldots, m$, the instance I' contains m_i such vertices $v_i^1, \ldots, v_i^{m_i}$, while, as noted above, an optimal solution of I opens at most m_i facilities at v_i.

Hence, instance I of Soft-CFLP reduces in $O(\sum_{i=1}^{m} m_i) = O(mB)$ time to an equivalent instance I' of RFLP, which by Proposition 3 is solvable in $O(|V(G')| \cdot B^{2\mathrm{tw}(G')}) = O((|V(G)| + mB) \cdot B^{2\mathrm{tw}(G)})$ time. To see that $\mathrm{tw}(G') = \mathrm{tw}(G)$, take a tree decomposition T of G and for each $v_i \in \mathcal{F}$

choose some bag $X_i \in T$ containing v_i and attach to X_i an edge leading to a new bag $X_{i,k} = \{v_i, v_i^k\}$ for each $k = 1, \dots, m_i$. This yields a valid tree decomposition T' of G'. Since every new bag has size 2, and any tree decomposition of a graph with edges must contain a bag of size at least 2, the maximum bag size remains unchanged. Thus, $\mathrm{tw}(G') = \mathrm{tw}(G)$. $\square$

5 Soft-UCFLP on a path

Paper [2] proposed a polynomial-time algorithm for the UCFLP with hard capacities on a path. The algorithm for the soft-capacitated case shown in this section relies on similar ideas with several modifications. First, dynamic programming subroutines in [2] iterate by the number of opened facilities, which in the hard-capacitated case must be at most m for any feasible solution. For the setting with soft capacities m is the number of facility locations, while the number of open facilities is at least $\sum_{j=1}^{n} b_j/a$ and can be non-polynomial in the input size, if there are client demands $b_j \gg a$. To fix that, we will show that for each client j with large demand $b_j \geq 3a$ most part of his demand can be served from a single location, and this location can be found in polynomial time in the preprocessing stage. Second, many dynamic programming subroutines from [2] become simpler in the soft-capacitated setting, since we no longer need to enforce that at most one facility is opened at each location; instead, opening an arbitrary number of facility copies at each location is allowed.

5.1. Properties of the optimal solution. Let $G = (V, E)$ be a path and assume that sets $V = \{1, \dots, |V|\}$, $\mathcal{F} = \{v_1, \dots, v_m\} \subseteq V$ and $\mathcal{C} = \{u_1, \dots, u_n\} \subseteq V$ are sorted in the order of traversing the path from left to right. Moreover, assume that the path is preprocessed, such that $V = \mathcal{F} \cup \mathcal{C}$. This can be done in linear time by sequentially replacing subpath $(v-1, v, v+1)$ with an edge $\{v-1, v+1\}$ of cost $c(\{v-1, v\}) + c(\{v, v+1\})$ for each $v \in V \setminus (\mathcal{F} \cup \mathcal{C})$. Now, $|V| \leq n + m$.

Following [2], for each $j' \leq j''$ by a client segment $[j', j'']$ we define a set of consecutive clients located at $\{u_{j'}, u_{j'+1}, \dots, u_{j''}\}$. Similarly, for $i' \leq i''$ by facility location segment $[i', i'']$ we define a set of consecutive facility locations $\{v_{i'}, v_{i'+1}, \dots, v_{i''}\} \subseteq \mathcal{F}$.

For the UCFLP with hard capacities on a path, where it is allowed to open *at most one* facility at each location, the following properties were shown in [2].

Proposition 4 ([2]). *The UCFLP-path with hard capacities has an optimal solution (y, x) satisfying:*

(1^0) *For any facilities $i', i'' \in I$ and clients $j', j'' \in J$ such that $i' < i''$ and $j' < j''$ it should be $x_{i'j''} = 0$, or $x_{i''j'} = 0$, or both.*

(2^0) *For any underutilized facilities i' and i'' facility segment $[i'; i'']$ contains open facilities number i^* and i^{**} such that they don't share a client and there are no open facilities in the segment $[i^* + 1; i^{**} - 1]$.*

From the property (1^0) it follows that each open facility serves a contiguous client segment $[j_\ell, j_r]$ and may share with other open facilities only clients on the boundaries of this segment, that is, clients j_ℓ and j_r. From properties (1^0) and (2^0) it follows that the set of all open facilities can be decomposed into non-intersecting facility segments, each containing at most one underutilized facility, and such that facility segments do not share clients with each other. These properties form the basis of the dynamic programming algorithm for the UCFLP-path with hard capacities in [2].

The algorithm for the Soft-UCFLP on a path will rely on weaker properties: we need to decompose only the client set into non-intersecting segments, while an optimal serving for each segment can be obtained independently.

Lemma 1. *Consider solution* (y, x) *for the Soft-UCFLP on a path, in which the set of clients can be partitioned into non-intersecting segments where for each such segment* $[j', j'']$*:*

- (0^*) *its demand is fully served by a subset* $\sigma_{j',j''}$ *of open facilities, and facilities in* $\sigma_{j',j''}$ *do not serve clients outside of* $[j', j'']$*;*
- (1^*) *for any facility locations* $i', i \in \{1, \dots, m\}$ *and clients* $j, k \in [j', j'']$ *with* $i' < i$ *and* $k < j$*, it is* $x_{i'j''} = 0$*, or* $x_{i''j'} = 0$*, or both;*
- (2^*) *there is at most one underutilized facility in* $\sigma_{j',j''}$*.*

Any such solution is feasible, and there exists an optimal solution for the Soft-UCFLP satisfying (0^*)–(2^*)*.*

Proof. First, note that a set of open facilities that satisfy (1)–(4) and (0^*) will form a feasible solution for Soft-UCFLP on a path, since it is allowed to open arbitrary number of facilities of the same type in each vertex of $\mathcal{F}$. Properties (1^*) and (2^*) don't violate the feasibility of the solution, either.

Now let $B = \sum_{j=1}^{n} b_j$ be the total client demand. Clearly, it is suboptimal to open more than $\lceil B/a \rceil$ facilities of type i for each $i = 1, \dots, m$, since the total capacity of $\lceil B/a \rceil$ facilities of type i is already enough to serve total client demand from location i. Consider the following straightforward pseudopolynomial reduction from an instance $\mathcal{I}$ of the Soft-UCFLP on a path to an equivalent instance $\mathcal{I}'$ of the UCFLP on a path with hard capacities: replace each facility location vertex v_i in $\mathcal{I}$ by a path $P_i = \{v_{i_1}, v_{i_2}, \dots, v_{i_{\lceil B/a \rceil}}\}$ with zero cost edges and $\lceil B/a \rceil$ vertices, in each of which at most one facility of type i is allowed to be open. By Proposition 4 there is an optimal solution (y', x') for $\mathcal{I}'$, where $y' = (y'_{i_k})$ and $x' = (x'_{i_k j})$, that satisfies properties (1^0) and (2^0), and thus, properties (0^*)–(2^*) are satisfied as well. Then, a solution (y, x) for $\mathcal{I}$ with $x_{ij} = \sum_{k=1}^{\lceil B/a \rceil} x'_{i_k j}$ and $y_i = \sum_{k=1}^{\lceil B/a \rceil} y'_{i_k}$, will have the same total cost and satisfy (0^*)–(2^*). □

Next, we show how to reduce the initial instance of the Soft-UCFLP on a path to an instance with bounded client demands, so that further in the dynamic programming routines we were able to safely iterate by the number of opened facilities.

Lemma 2. *Soft-UCFLP on a path admits an optimal solution in which for each client j with demand $b_j \geq 2a$ there is at most one location i, that sends $x_{ij} \geq a$ product to j.*

Proof. Consider an optimal solution satisfying (1*) in which some client j with $b_j \geq 2a$ gets $x_{i_1,j} = a \cdot m_1 + r_1$ product from location i_1 and $x_{i_2,j} = a \cdot m_2 + r_2$ product units from location i_2, where $0 \leq r_1, r_2 < a, \;\; m_1, m_2 \geq 1$, and $r_1, r_2, m_1, m_2 \in \mathbb{N} \cup \{0\}$. W.l.o.g, we can assume that there are m_1 fully utilized facilities at i_1 and m_2 fully utilized facilities at i_2, that send their full capacity to j. Let $i^* = \arg\min\{f_i + a \cdot d(v_i, v_j) \mid i = i_1, i_2\}$, where f_i is the cost of opening a facility at location i. Then replacing each open facility at $\{i_1, i_2\} \setminus \{i^*\}$ that sends its full capacity to the client j by a new fully utilized facility at i^* is feasible, doesn't violate (1*), doesn't affect the service of other clients, and doesn't increase the overall cost of the solution. □

Lemma 3. *Let $\alpha_1, \ldots, \alpha_k \in \mathbb{N}$ satisfy $\alpha_1 + \cdots + \alpha_k \leq a$. Suppose that, in an feasible solution to Soft-UCFLP, one facility serves exactly α_p units of demand of each client j_p, $p = 1, \ldots, k$, and does not serve any other client. Then the optimal location of this facility can be chosen independently of the rest of the solution as*

$$i^* = \arg\min_{1 \leq i \leq m} \{f_i + \sum_{p=1}^{k} \alpha_k d(v_i, u_{j_p})\}. \tag{7}$$

Proof. By definition, such facility have enough capacity to serve mentioned demands, and does not serve any other demand (if $\alpha_1 + \cdots + \alpha_k = a$ it cannot serve any other demand). Moreover, its location choice does not affect the rest of the solution, since an arbitrary number of copies of each facility can be opened. So, the location can be chosen independently, and the optimal choice is given by (7). □

Lemma 4. *Given an instance of the Soft-UCFLP on a path, one can in polynomial time reduce initial client demands $b_j, j = 1, \ldots, n$, to*

$$b'_j = \min\{b_j, 2a + (b_j \bmod a)\},$$

and get an equivalent instance of the problem.

Proof. Consider an optimal solution for the the Soft-UCFLP on a path satisfying Lemma 2 and (1*). Let client j with $b_j \geq 3a$ in this solution be jointly served from $p \geq 2$ locations $i_1 < i_2 < \cdots < i_p$. By property (1*), facilities at locations $i_2, \ldots, i_{p-1}$ can only serve client j. By Lemma 2, all fully utilized facilities serving only j should be open at one location. By Lemma 3, this location can be found in polynomial time as

$$i^{(j)} = \arg\min_{1 \leq i \leq m} \{f_i + a \cdot d(v_i, u_j)\}. \tag{8}$$

Let $a \cdot s$ be the total amount of product j gets from these fully utilized facilities. By Proposition 1, there can also be at most one underutilized facility that only serves j, let q be the amount of product it sends to j.

Finally, at each of i_1 and i_p there can be one facility that serves not only j, let $0 < \alpha_1 < a$ and $0 < \alpha_p < a$ be the amount of product j receives from these facilities. From $b_j = \alpha_1 + \alpha_p + q + a \cdot s$ we get

$$s = b_j/a - \big(\alpha_1 + \alpha_p + q\big)/a \geq \lfloor b_j/a \rfloor - 2,$$

since $\alpha_1, \alpha_p, q < a$ and s must be an integer. Therefore, it is always safe to serve at least $a(\lfloor b_j/a \rfloor - 2)$ units of j-th demand from $i^{(j)}$, paying constant $(\lfloor b_j/a \rfloor - 2)(f_{i^{(j)}} + a \cdot d(v_{i^{(j)}}, u_j))$ in the goal function, and leaving j-th client with the reduced demand $b'_j = 2a + (b_j \bmod a)$. □

5.2. Dynamic programming algorithm. In this section we consider an instance of Soft-UCFLP on a path after applying Lemma 4, that is, with all client demands being less than $3a$.

Let $R(j)$ be the cost of an optimal solution satisfying (0^*)–(2^*) in a sub-problem with the set of clients $\{1, \dots, j\}$ and the full set of facility locations $\{1, \dots, m\}$. Let $Q(j', j)$ be the cost of an optimal solution in a subproblem $\Pi_Q[j', j]$ with the set of clients $[j', j]$ and facility locations $\{1, \dots, m\}$, such that $[j', j]$ is served by the best possible facility subset $\sigma_{j',j}$ containing at most one underutilized facility. Then, by Lemma 1, for each $j = 1, \dots, n$:

$$R(j) = \min_{1 \leq j' \leq j} \{R(j'-1) + Q(j', j)\}, \quad R(0) = 0, \tag{9}$$

and $R(n)$ is the cost of the optimal solution of the Soft-UCFLP-path with n clients, m facility locations and bounded client demands.

The rest of this subsection is devoted to deriving recurrence relations to compute values $Q(j', j'')$.

5.2.1. Solving the subproblem $\Pi_Q[j', j'']$. Let $B_j = \sum_{k=1}^{j} b_k$ be the total demand of the first j clients, $j \in \{1, \dots, n\}$, with $B_0 = 0$; $B_{k,j} = B_j - B_{k-1}$ be the total demand of clients from $[k, j]$, $1 \leq k \leq j \leq n$, in particular, $B_{j,j} = b_j$, and $B_{k,j} = 0$ if $j < k$; $C(i, j) = \sum_{k=1}^{j} d(v_i, u_k) \cdot b_k$ be the transportation cost of serving all clients in $[1, j]$ from location $i \in \{1, \dots, m\}$, with $C(i, 0) = 0$; $r_{j'j''} = \lfloor B_{j',j''}/a \rfloor$ and $q_{j'j''} = B_{j',j''} - a \cdot r_{j',j''}$.

By definition, any feasible solution to $\Pi_Q[j', j'']$ contains at most one underutilized facility. Hence, to serve the full demand $B_{j',j''}$ of the client segment $[j', j'']$ one must open $r_{j'j''}$ fully utilized facilities and, if $q_{j'j''} > 0$, one additional underutilized facility delivering exactly $q_{j'j''} < a$ units of product. Note that $q_{j'j''}$ can be non-polynomial in the size of the input, yet, since the client demands are bounded by Lemma 4, $r_{j'j''} \leq B_{j',j''}/a < 3(j'' - j') \leq 3n$.

By Lemma 1, there is an optimal solution to $\Pi_Q[j', j'']$ where property (1^*) is satisfied, that is, client serving is sequential in the path order. Thus, there exists $t \in \{0, \dots, r_{j'j''}\}$ such that the left block of $a \cdot t$ units of total client demand of $[j', j'']$ is served by a set of t fully utilized facilities, the next $q_{j'j''}$ units by one underutilized facility, if $q_{j'j''} > 0$, and the remaining right block of $a(r_{j'j''} - t)$ demand units is served by another set of $r_{j'j''} - t$ fully utilized

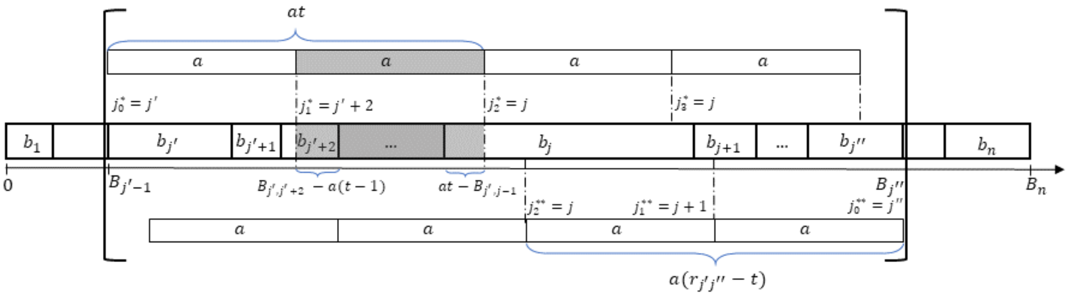

FIG. 2. Serving the client segment $[j', j'']$ in $\Pi_Q[j', j'']$. Rectangles in the middle stand for client demands. Rectangles on top (bottom) represent the service intervals of the left (right) fully utilized facilities. The multiset of clients $T^L_{j'} = \{j^*_0 = j', j^*_1 = j' + 2, j^*_2 = j, j^*_3 = j\}$ and $T^R_{j''} = \{j^{**}_0 = j'', j^{**}_1 = j + 1, j^{**}_2 = j, j^*_3 = j' + 2\}$.

facilities, covering client demands in the path order. Equivalently, consider the cumulative-demand interval $[B_{j'-1}, B_{j''}]$ of length exactly $B_{j',j''}$ (see Fig. 2) that corresponds to the total client demand of $[j', j'']$ and should be covered with $r_{j'j''}$ full blocks of length a and one residual block of length $q_{j'j''}$. The residual block, thus, can occupy one of $r_{j'j''} + 1$ possible positions: $[B_{j'-1} + a \cdot t, B_{j'-1} + a \cdot t + q_{j'j''}]$, $t = 0, \ldots, r_{j'j''}$.

From this we construct the dynamic programming routine for $Q(j', j'')$. Note that, for a fixed t, each fully utilized facility described above serve a separate continuous demand block of size a and nothing else, while the underutilized facility serves exactly the chosen residual block of demand $q_{j',j''}$ and nothing else. Thus, all these facilities satisfy Lemma 3 and their optimal locations can be found independently.

For each $t = 0, \ldots r_{j'j''}$ let $G^L_{j'}(t)$ be the minimum cost of serving $[B_{j'-1},\ B_{j'-1} + a \cdot t]$ by t fully utilized facilities, $S_{j',j''}(t)$ be the minimum cost of serving $[B_{j'-1} + a \cdot t,\ B_{j'-1} + a \cdot t + q_{j',j''}]$ by one underutilized facility, and $G^R_{j''}(t)$ be the minimum cost of serving $[B_{j''} - (r_{j',j''} - t)a,\ B_{j''}]$ by $r_{j',j''} - t$ fully utilized facilities. Then

$$Q(j', j'') = \min_{0 \le t \le r_{j'j''}} \{G^L_{j'}(t) + S_{j'j''}(t) + G^R_{j''}(t)\}. \tag{10}$$

Optimally serving the left block. Following [2], we introduce a multiset of clients $T^L_{j'} = \{j^*_0, \ldots, j^*_{r'}\}$, such that $r' = \lfloor B_{j',n}/a \rfloor < 3(n - j')$, each $j^*_p \in \{j', \ldots, n\}$, $j^*_0 = j'$ and

$$B_{j',j^*_p - 1} < pa \le B_{j',j^*_p}, \quad 1 \le p \le r'.$$

Since each reduced client demand satisfies $b_j < 3a$, any client $j \in \{j', \ldots, n\}$ may appear in $T^L_{j'}$ at most three times.

The p-th full block $[B_{j'-1} + a(p-1), B_{j'-1} + ap]$ of length a contains the full demands of client segment $[j^*_{p-1} + 1, j^*_p - 1]$ together with a suffix $(B_{j',j^*_{p-1}} - (p-1)a)$ of j^*_{p-1}-th client demand and a prefix $(pa - B_{j',j^*_p-1})$ of j^*_p-th client demand; see Fig 2. Let $g^L_{j'}(i,p)$ be the cost of serving the p-th full block by one fully utilized facility located at i. Then

$$\begin{gathered} g^L_{j'}(i,p) = f_i + C(i, j^*_p - 1) - C(i, j^*_{p-1}) + \\ \big(B_{j',j^*_{p-1}} - (p-1)a\big) d(v_i, u_{j^*_{p-1}}) + \big(pa - B_{j',j^*_p-1}\big) d(v_i, u_{j^*_p}). \end{gathered} \tag{11}$$

In case $j^*_{p-1} = j^*_p$, it is simply $g^L_{j'}(i,p) = f_i + a \cdot d(v_i, u_{j^*_p})$, but formula (11) remains valid. By Lemma 3, the minimum cost of serving this block can be found independently as

$$g^L_{j'}(p) = \min_{1 \le i \le m} g^L_{j'}(i,p), \tag{12}$$

and the minimum cost of serving the first t full blocks by a set of fully-utilized facilities is simply

$$G^L_{j'}(0) = 0, \quad G^L_{j'}(t) = \sum_{p=1}^{t} g^L_{pj'} = G^L_{j'}(t-1) + g^L_{j'}(t) \quad t = 1, \ldots, r'. \tag{13}$$

Optimally serving the right block. Similarly, for the family of right subproblems, fix j'' and consider the multiset $T^R_{j''} = \{j^{**}_0, \ldots, j^{**}_{r''}\}$, where $r'' = \left\lfloor \frac{B_{j''}}{a} \right\rfloor < 3j''$ and each $j^{**}_p \in \{1, \ldots, j''\}$ is defined by $j^{**}_0 := j''$ and

$$B_{j^{**}_p, j''} < pa \le B_{j^{**}_{p-1}, j''}, \qquad p = 1, \ldots, r''.$$

The p-th full block from the right, $[B_{j''} - ap,\ B_{j''} - a(p-1)]$, contains the full demands of clients $[j^{**}_p + 1,\ j^{**}_{p-1} - 1]$, together with a suffix $pa - B_{j^{**}_p+1, j''}$ of the demand of client j^{**}_p and a prefix $B_{j^{**}_{p-1}, j''} - (p-1)a$ of the demand of client j^{**}_{p-1}. Let $g^R_{j''}(p)$ be the minimum cost of serving this block. Then

$$\begin{gathered} g^R_{j''}(p) = \min_{1 \le i \le m} \{ f_i + C(i, j^{**}_{p-1} - 1) - C(i, j^{**}_p) + \\ \big(ta - B_{j^{**}_p+1, j''}\big) d(v_i, u_{j^{**}_p}) + \big(B_{j^{**}_{p-1}, j''} - (p-1)a\big) d(v_i, u_{j^{**}_{p-1}}) \}, \end{gathered} \tag{14}$$

and the minimum cost of serving the first $r'' - t$ full blocks form the right is

$$G^R_{j''}(t) = \sum_{p=t+1}^{r''} g^R_{j''}(p) = G^R_{j''}(t+1) + g^R_{j''}(t+1), \quad t = r'' - 1, \ldots, 0, \tag{15}$$

with $G^R_{j''}(r'') = 0$.

Optimally serving the residual block of length $q_{j'j''}$. Finally, the optimal cost of serving segment $[B_{j'-1} + at, B_{j'-1} + at + q_{j'j''}]$ from the underutilized facility is

$$\begin{aligned} S_{j'j''}(t) = \min_{1\le i\le m}\Big\{ f_i + \big(C(i, j^{**}_{r-t}-1) - C(i, j^{*}_{t-1})\big) + \\ \big(B_{j',j^{*}_{t-1}} - (t-1)a\big) d(v_i, u_{j^{*}_{t-1}}) + \big(B_{j^{**}_{r-t},j''} - (r-t)a\big) d(v_i, u_{j^{**}_{r-t}})\Big\}, \end{aligned} \tag{16}$$

where $r = r_{j'j''} = \lfloor B_{j'j''}/a \rfloor \le 3(j''-j') \le 3n$.

5.3. Time complexity. We start by proving a technical lemma that allows to reduce the running time of several dynamic programming subroutines (like (8), (11)-(12), (14), and (16)) by one order of magnitude.

Lemma 5. *Assuming that all values $d(u,v)$ and $C(i,j)$ are accessible in $O(1)$ time, one can compute the following in total $O(m+n)$ time*

(i) *the values $i^{(j)} := \arg\min_{1\le i\le m}\{f_i + a\cdot d(v_i,u_j)\}$ for all $j = 1,\dots,n$.*

(ii) *the values $H(t) := \min_{1\le i\le m} H(i,t)$ for all $t = 1,\dots,|T|$, where T and $H(i,t)$ are defined as follows. The set T is a sorted (multi)set of clients, such that $|T| = O(n)$; for each $t\in\{1,\dots,|T|\}$, the values $\alpha_1(t)$ and $\alpha_2(t)$ are constants accessible in $O(1)$ time and satisfy $\sum_{j=j_t+1}^{j_{t+1}-1} b_j + \alpha_1(t) + \alpha_2(t) = \gamma$ where γ is a fixed constant, $j_t, j_{t+1}\in T$, and*

$$H(i,t) := f_i + C(i, j_{t+1}-1) - C(i,j_t) + d(v_i,u_{j_t})\cdot\alpha_1(t) + d(v_i,u_{j_{t+1}})\cdot\alpha_2(t).$$

Proof. For a fixed parameter γ, define the functions $F^-(j,\gamma)$ and $F^+(j,\gamma)$ as

$$F^-(j,\gamma) := \min_{i: v_i\le u_j}\{f_i - \gamma\cdot d(1,v_i)\} \text{ and } F^+(j,\gamma) := \min_{i: v_i\ge u_j}\{f_i + \gamma\cdot d(1,v_i)\}.$$

These values satisfy the recurrences

$$F^-(j,\gamma) = \min\Big\{F^-(j-1,\gamma), \min_{i: u_{j-1}<v_i\le u_j}\{f_i - \gamma\cdot d(1,v_i)\}\Big\},\ j = 1,\dots,n,$$

$$F^+(j,\gamma) = \min\Big\{F^+(j+1,\gamma), \min_{i: u_{j}\le v_i< u_{j+1}}\{f_i + \gamma\cdot d(1,v_i)\}\Big\},\ j = n,\dots,1,$$

with boundary values

$$F^-(1,\gamma) = \min_{i: v_i\le u_1}\{f_i - \gamma d(1,v_i)\} \text{ and } F^+(n,\gamma) = \min_{i: v_i\ge u_n}\{f_i + \gamma d(1,v_i)\}.$$

From the recurrence above, for each next $j = 1,\dots,n$ computing $F^-(j,\gamma)$ together with its minimizer takes $O(u_j - u_{j-1})$ time. Thus, all these values can be computed in total $O(|V|) = O(n+m)$ time. Similarly, all values $F^+(j,\gamma)$ and their minimizers can be computed in total $O(n+m)$ time.

Now (i) follows from

$$\begin{aligned} i^{(j)} = \arg\min_i\Big\{ \min_{i: v_i\le u_j}\{f_i - a\cdot d(1,v_i) + a\cdot d(1,u_j)\}, \\ \min_{i: v_i\ge u_j}\{f_i + a\cdot d(1,v_i) - a\cdot d(1,u_j)\}\Big\} = \arg\min_i\{F^-(j,a), F^+(j,a)\}. \end{aligned}$$

Thus, having all values $F^-(j,a)$ and $F^+(j,a)$ computed in $O(m+n)$ time, each $i^{(j)}$ can be obtained in $O(1)$ time, and computing all $i^{(j)}$ takes $O(n)$ time.

To show (ii) we use the same idea. First, let

$$h_t := \sum_{j=j_t+1}^{j_{t+1}-1} b_j d(u_j,1) + \alpha_1(t) d(u_{j_t},1) + \alpha_2(t) d(u_{j_{t+1}},1) = \\ C(1,u_{j_{t+1}-1}) - C(1,u_{j_t}) + \alpha_1(t) d(u_{j_t},1) + \alpha_2(t) d(u_{j_{t+1}},1).$$

Each h_t don't depend on i and can be obtained in $O(1)$ time. Since by assumption $\sum_{j=j_t+1}^{j_{t+1}-1} b_j + \alpha_1(t) + \alpha_2(t) = \gamma$, we have

$$H(t) = \min_{1\le i\le m} \{f_i + \sum_{j=j_t+1}^{j_{t+1}-1} b_j \cdot |d(1,v_i) - d(1,u_j)| + \alpha_1(t) \cdot |d(1,v_i) - d(1,u_{j_t})| + \\ \alpha_2(t) \cdot |d(1,v_i) - d(1,u_{j_{t+1}})|\} = \min\Big\{ \min_{i:v_i\le u_{j_t}} \{f_i - \gamma d(1,v_i)\} + h_t, \\ \min_{i:v_i\ge u_{j_{t+1}}} \{f_i + \gamma d(1,v_i)\} - h_t, \min_{i:u_{j_t}<v_i<u_{j_{t+1}}} H(i,t)\Big\} = \\ \min\Big\{F^-(j_t,\gamma) + h_t, F^+(j_{t+1},\gamma) - h_t, \min_{i:u_{j_t}<v_i<u_{j_{t+1}}} H(i,t)\Big\}.$$

For a fixed t, the first two terms in the last expression can be obtained in $O(1)$ time, while the third requires inspecting only facilities $i : u_{j_t} < v_i < u_{j_{t+1}}$, which takes $O(u_{j_{t+1}} - u_{j_t})$ time, since for a fixed pair (i,t) it takes $O(1)$ time to compute $H(i,t)$. Thus, all values $H(t)$ can be found in total $O(n+m)$ time. □

Theorem 3. *An optimal solution to the Soft-UCFLP on a path can be found in $O(n^3 + n^2 m)$ time.*

Proof. The algorithm and its running time can be summarized as follows. First, compute the prefix distances along the path recursively as $d(1,v) = d(1,v-1) + c(\{v-1,v\})$ for all vertices $v \in V$. This takes total time $T_1 = O(|V|) = O(n+m)$. Now, if necessary, for any two vertices $u, v \in V$, $d(u,v) = |d(1,v) - d(1,u)|$ can be obtained in $O(1)$ time. Next, reduce client demands as in Lemma 4. Namely, for each client $j = 1,\dots,n$ with demand at least $3a$ find the best location $i^{(j)}$ defined by (8) to serve certain multiple-of-a part of its demand leaving the remaining bounded demand $b_j < 3a$. By Lemma 5(i) this step takes total time $T_2 = O(n+m)$.

In the reduced instance, compute prefix sums $B_j = B_{j-1} + b_j$ in total time $T_3 = O(n)$ and $C(i,j) = C(i,j-1) + d(v_i,u_j) \cdot b_j$ in total time $T_4 = O(nm)$. Once the values B_j are known, if necessary, any $B_{j'j} = B_j - B_{j'-1}$ can be obtained in $O(1)$ time.

The cost of an optimal solution to the reduced instance of Soft-UCFLP on a path is then computed by dynamic programming using relations (9). Once all values $Q(j',j'')$ are available, (9) can be computed in total time

$T_5 = O(n^2)$. For each fixed pair $1 \le j' \le j'' \le n$, the value $Q(j', j'')$ is computed by recurrence (10), which takes $O(j'' - j') = O(n)$ time once its terms $G^L_{j'}(t)$, $S_{j'j''}(t)$ and $G^R_{j''}(t)$ are known. Hence, computing all values $Q(j', j'')$ takes $T_6 = O(n^3)$ time.

For a fixed $j' \in \{1, \dots, n\}$, computing all values $G^L_{j'}(t)$ for $t = 0, \dots, |T^L_j|$ takes in total $O(n + m)$ time: constructing $T^L_{j'}$, where $|T^L_{j'}| \le 3(n - j')$, takes $O(n)$ time; computing all values $g^L_{j'}(t)$ defined by (11)-(12) takes total $O(n+m)$ time by Lemma 5(ii); and then calculating $G^L_{j'}(t)$ according to (13) for each t takes $O(1)$ time. Therefore, computing all values $G^L_{j'}(t)$ for all j', t takes total time $T_7 = O(n(n + m))$. Similarly, all values $G^R_{j''}(t)$ can be computed in total $T_8 = O(n(n + m))$ time.

Finally, for each fixed pair $1 \le j' \le j'' \le n$, all values $S_{j'j''}(t)$, defined by (16), for $t = 0, \dots, r_{j'j''}$ can be computed in total $O(n + m)$ time by Lemma 5(ii). So, computing $S_{j'j''}(t)$ for all j', j'', t requires $T_9 = O(n^3+mn^2)$ time in total.

Overall, the running-time of the algorithm is $\sum_{i=1}^{9} T_i = O(n^3 + n^2 m)$. □

6 Conclusion

In this paper, we studied the network multiple-allocation Soft-Capacitated Facility Location Problem on tree-like graphs. We proved that the Soft-UCFLP remains NP-hard on trees, showing that, with respect to polynomial-time solvability on trees and paths, the soft-capacitated setting is not much simpler than the hard-capacitated one. On the positive side, we obtained an $O(n^3 + n^2 m)$-time algorithm for the Soft-UCFLP on paths, where n is the number of clients and m is the number of facility locations. We also showed that Soft-CFLP admits a pseudopolynomial-time algorithm on graphs of bounded treewidth.

Theorem 1 in fact implies NP-hardness of the Soft-UCFLP on trees of pathwidth at least 2, since the tree constructed in the reduction has pathwidth 2. Graphs of pathwidth at most 1 are precisely caterpillar trees, a class that includes paths and stars. For the hard-capacitated UCFLP, NP-hardness on stars was shown in [4]; however, that reduction does not appear to extend to the soft-capacitated setting. So, the complexity of the Soft-UCFLP on stars, and more general caterpillar trees, remains open. Another natural direction for the future work is improving the running time of the path algorithm, or proving lower bounds on the running time achievable for this case.

References

[1] A. Ageev *A criterion of polynomial-time solvability for the network location problem*, Integer programming and combinatorial optimization, Proc. IPCO II Conf, (1992), 237–245.

[2] A. A. Ageev, E. K. Gimadi, A. A. Kurochkin, *Polynomial-time algorithm for the Capacitated Facility Location Problem with uniform capacities on a path (In Russian)*, Discretnyi analiz i issledovanie operacii, **16** (2009), 3–18.

[3] A. Ageev, E. Gimadi, A. Shtepa, *How fast can the uniform capacitated facility location problem be solved on path graphs*, Lecture Notes in Computer Science, **13217** (2022), 303–314.

[4] A. A. Ageev, E. K. Gimadi, O. Y. Tsidulko, A. A. Shtepa, *Capacitated facility location problem on tree-like graphs (In Russian)*, Trudy Inst. Mat. i Mekh. UrO RAN, **28** (2022), 24–44.

[5] A. Aggarwal, A. Louis, M. Bansal, N. Garg, N. Gupta, S. Gupta, S. Jain, *A 3-approximation algorithm for the facility location problem with uniform capacities*, Mathematical Programming, **141** (2013), 527–547.

[6] M. Bansal, N. Garg, N. Gupta, *A 5-approximation for capacitated facility location*, Lecture Notes in Computer Science, **7501** (2012), 133–144.

[7] H. L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth*, Theoretical Computer Science, **209** (1998), 1–45.

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

[9] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., USA, 1990.

[10] E. K. Gimadi, O. Y. Tsidulko, *On some efficiently solvable classes of the network facility location problem with constraints on the capacities of communication lines*, Proceedings of the Steklov Institute of Mathematics, **313** (2021), 58–72.

[11] D. Granot, D. Skorin-Kapov, *On some optimization problems on k-trees and partial k-trees*, Discrete Applied Mathematics, **48** (1994) 129–145.

[12] G. Laporte, S. Nickel, F. S. da Gama, *Location science*, Switzerland: Springer Internat. Publ., 2015.

[13] M. Mahdian, M. Pál, *Universal facility location*, Lecture Notes in Computer Science, **2832** (2003), 409–421.

[14] M. Mahdian, Y. Ye, J. Zhang, *Approximation algorithms for metric facility location problems*, SIAM Journal on Computing, **36** (2006), 411–432.

[15] R. Hassin, A. Tamir, *Improved complexity bounds for location problems on the real line*, Operations Research Letters, **10** (1991), 395–402.

[16] R. Shah, V. Farach-Colton, *Undiscretized dynamic programming: faster algorithms for facility location and related problems on trees*, Proc. of the 13th Ann. ACM-SIAM Symp. on Discrete Algorithms, SODA, (2002), 108–115.

[17] H. Xiao, J. Zhang, Z. Zhang, W. Li, *A survey of approximation algorithms for the universal facility location problem*, Mathematics **13**:7:1023 (2025).

Appendix: Error in the algorithm for the UCFLP on a path in papers [3, 4]

The conference paper [3], whose results were later included in the journal paper [4], attempts to improve the time complexity of the algorithm for the UCFLP with hard capacities on a path proposed in [2] from the initial $O(m^4n^2)$ to $O(m^2n^2)$. The proposed improvements relied on rearranging the minima within recurrences and precomputing certain terms individually prior to computing the global minima. However, these steps were found to contain certain errors.

From Proposition 4 proved in [2], it follows that the optimal solution for the hard-capacitated UCFLP on a path can be decomposed into subproblems where each separate client segment $[j', j'']$ should be fully served by the facilities located in a separate segment $[i', i'']$ with at most one underutilized facility. To find the cost $Q_{i',i''}(j', j'')$ of the optimal solution for such subproblem with fixed $1 \le i' \le i'' \le m$, $1 \le j' \le j'' \le n$, paper [2] showed how to compute the minimum cost

- S_{it} of serving the demand block $[B_{j'-1} + a \cdot t,\ B_{j'-1} + a \cdot t + q_{j',j''}]$ by one underutilized facility located at i;
- $\widetilde{G}^L_{i,t,j'}$ of serving the left demand block $[B_{j'-1},\ B_{j'-1} + a \cdot t]$ by t fully utilized facilities located in $[i', i]$, $i \ge t$;
- $\widetilde{G}^R_{i,t,j''}$ of serving the right block $[B_{j'-1} + a \cdot t + q_{j',j''},\ B_{j''}]$ by $r_{j',j''} - t$ fully utilized facilities located in $[i, i'']$, $i'' - i > r_{j',j''} - t$.

Not that, although in [2] the corresponding indices were omitted, in the hard-capacitated UCFLP, values $\widetilde{G}^L_{i,t,j'}$ depend on i', and values $\widetilde{G}^R_{i,t,j''}$ depend on i'' by definition. Then the cost $Q_{i',i''}(j', j'')$ of the optimal solution to the subproblem was computed as

$$Q_{i',i''}(j', j'') = \min_{i' \le i \le i''} \min_{0 \le t \le r_{j'j''}} \{\widetilde{G}^L_{i-1,t,j'} + S_{it} + \widetilde{G}^R_{i+1,t,j''}\},$$

which, once all of the terms in the minimum are known, can be found in $O(m^2)$ time for fixed i', i'', j', j'', since $r_{j'j''} \le m$ in the hard-capacitated case. So, finding all $Q_{i',i''}(j', j'')$ takes a total of $O(m^4 n^2)$ time, and, being the most time-consuming part, it determines the time complexity of the whole algorithm in [2].

To compute $Q_{i',i''}(j', j'')$ faster, equation (8) in [3] and equation (4.4) in [4] introduced values $Q_i(j', j'')$

$$Q_i(j', j'') := \min_{0 \le t \le r_{j'j''}} \{\widetilde{G}^L_{i-1,t,j'} + S_{it} + \widetilde{G}^R_{i+1,t,j''}\},$$

which were supposed to be computed separately, and were claimed to be found in total $O(m^2 n^2)$ time for all $1 \le i \le m$, $1 \le j' < j'' \le n$, taking into account $r_{j'j''} \le m$. Yet, since in fact $\widetilde{G}^L_{i,t,j'}$ depends on i' and $\widetilde{G}^R_{i,t,j''}$ depends on i'', the values $Q_i(j', j'')$ should also depend on i' and i''. Therefore, the definition above is ill-defined. If defined correctly as $Q_{i',i'',i}(j', j'')$, the overall time for computing all these values would already be $O(m^4 n^2)$, and an algorithm using these values cannot be faster than the algorithm from [2].

Oxana Yurievna Tsidulko
Sobolev Institute of Mathematics,
pr. Koptyuga, 4,
630090, Novosibirsk, Russia
Email address: `tsidulko@math.nsc.ru`