Facility Location with Client Latencies: Linear Programming based Techniques for Minimum Latency Problems *

Deeparnab Chakrabarty[1] and Chaitanya Swamy[2]

[1] Dept. of Comp. and Inf. Science, Univ. of Pennsylvania, PA 19104
deepc@seas.upenn.edu
[2] Combinatorics and Optimization, Univ. Waterloo, Waterloo, ON N2L 3G1.
cswamy@math.uwaterloo.ca

Abstract. We introduce a problem that is a common generalization of the uncapacitated facility location and minimum latency (ML) problems, where facilities need to be opened to serve clients and also need to be sequentially activated before they can provide service. Formally, we are given a set $\mathcal{F}$ of n facilities with facility-opening costs f_i, a set $\mathcal{D}$ of m clients, connection costs c_{ij} specifying the cost of assigning a client j to a facility i, a root node r denoting the depot, and a time metric d on $\mathcal{F} \cup \{r\}$. Our goal is to open a subset F of facilities, find a path P starting at r and spanning F to activate the open facilities, and connect each client j to a facility $\phi(j) \in F$, so as to minimize $\sum_{i \in F} f_i + \sum_{clients\, j} (c_{\phi(j),j} + t_j)$, where t_j is the time taken to reach $\phi(j)$ along path P. We call this the *minimum latency uncapacitated facility location* (MLUFL) problem.
Our main result is an $O(\log n \cdot \max(\log n, \log m))$-approximation for MLUFL. We also show that any improvement in this approximation guarantee, implies an improvement in the (current-best) approximation factor for group Steiner tree. We obtain *constant* approximations for two natural special cases of the problem: (a) related MLUFL (metric connection costs that are a scalar multiple of the time metric); (b) metric uniform MLUFL (metric connection costs, uniform time-metric). Our LP-based methods are versatile and easily adapted to yield approximation guarantees for MLUFL in various more general settings, such as (i) when the latency-cost of a client is a function of the delay faced by the facility to which it is connected; and (ii) the k-route version, where k vehicles are routed in parallel to activate the open facilities. Our LP-based understanding of MLUFL also offers some LP-based insights into ML, which we believe is a promising direction for obtaining improvements for ML.

1 Introduction

Facility location and vehicle routing problems are two broad classes of combinatorial optimization problems that have been widely studied in the Operations Research community (see, e.g., [15, 19]), and have a wide range of applications. Both problems can be described in terms of an

* A full version [4] is available on the CS arXiv

underlying set of clients that need to be serviced. In facility location problems, there is a candidate set of facilities that provide service, and the goal is to open some facilities and connect each client to an open facility so as to minimize some combination of the facility-opening and client-connection costs. Vehicle routing problems consider the setting where a vehicle (delivery-man or repairman) provides service, and the goal is to plan a route that visits (and hence services) the clients as quickly as possible. Two common objectives considered are: (i) minimize the total length of the vehicle's route, giving rise to the traveling salesman problem (TSP), and (ii) (adopting a client-oriented approach) minimize the sum of the client delays, giving rise to minimum latency (ML) problems.

These two classes of problems have mostly been considered separately. However, various logistics problems involve both facility-location and vehicle-routing components. For example, consider the following oft-cited prototypical example of a facility location problem: a company wants to determine where to open its retail outlets so as to serve its customers effectively. Now, inventory at the outlets needs to be replenished or ordered (e.g., from a depot); naturally, a customer cannot be served by an outlet unless the outlet has the inventory demanded by it, and delays incurred in procuring inventory might adversely impact customers. Hence, it makes sense for the company to *also* keep in mind the latencies faced by the customers while making its decisions about where to open outlets, which clients to serve at each outlet, and in what order to replenish the open outlets, thereby adding a vehicle-routing component to the problem.

We propose a mathematical model that is a common generalization of the *uncapacitated facility location* (UFL) and *minimum latency* (ML) problems, and abstracts such settings where facilities need to be "*activated*" before they can provide service. Formally, as in UFL, we have a set $\mathcal{F}$ of n facilities, and a set $\mathcal{D}$ of m clients. Opening facility i incurs a *facility-opening cost* f_i, and assigning a client j to a facility i incurs *connection cost* c_{ij}. (The c_{ij}s need not form a metric.) Taking a lead from minimum latency problems, we model activation delays as follows. We have a root (depot) node r, and a *time metric* d on $\mathcal{F} \cup \{r\}$. A feasible solution specifies a subset $F \subseteq \mathcal{F}$ of facilities to open, a path P starting at r and spanning F along which the open facilities are activated, and assigns each client j to an open facility $\phi(j) \in F$. The cost of such a solution is

$$\sum_{i \in F} f_i + \sum_{j \in \mathcal{D}} \left(c_{\phi(j)j} + t_j \right) \tag{1}$$

where $t_j = d_P(r, \phi(j))$ is the time taken to reach facility $\phi(j)$ along path P. One can view c_{ij} as the time facility i takes to serve client j after it

has been activated, in which case $(c_{\phi(j),j} + t_j)$ is the delay faced by client j. (Alternatively, if c_{ij} denotes the time taken by a client j to travel to facility i, then the delay faced by j is $\max(c_{\phi(j),j}, t_j)$, which is within a factor 2 of the sum.) We refer to t_j as client j's latency cost. The goal is to find a solution with minimum total cost. We call this the *minimum-latency uncapacitated facility location* (MLUFL) problem.

Apart from being a natural problem of interest, we find MLUFL appealing since it generalizes various diverse problems of interest, in addition to UFL and ML. One such problem, which captures much of the combinatorial core of MLUFL, is what we call the *minimum group latency* (MGL) problem: given an undirected graph with metric edge weights $\{d_e\}$, groups $\{G_j\}$ of vertices, and a root r, the goal is to find a path starting at r that minimizes the sum of the cover times of the groups, where the cover time of G_j is the first time at which some $i \in G_j$ is visited on the path. Observe that MGL can be cast as MLUFL with zero facility costs (where $\mathcal{F} =$ node-set $\setminus \{r\}$), where for each group G_j, we create a client j with $c_{ij} = 0$ if $i \in G_j$ and ∞ otherwise. Note that we may assume that the groups are disjoint (by creating multiple co-located copies of a node), in which case these c_{ij}s form a metric. MGL itself captures various other problems. Clearly, when each G_j is a singleton, we obtain the minimum latency problem. Given a set-cover instance, if we consider a graph whose nodes are (r and) the sets, we create a group G_j for each element j consisting of the sets containing it, and consider the uniform metric, then this MGL problem is simply the *min-sum set cover* (MSSC) problem [9].

Our results and techniques. Our main result is an $O\big(\log n \cdot \max(\log m, \log n)\big)$-approximation algorithm for MLUFL (Section 2.1), which for the special case of MGL, implies an $O(\log^2 n)$ approximation. Complementing this, we show that a ρ-approximation algorithm for MGL yields an $O(\rho \log m)$-approximation algorithm for the *group Steiner tree* (GST) problem [10] on n nodes and m groups. So an improved approximation ratio for MLUFL would yield a corresponding improvement for GST, whose approximation ratio has remained at $O(\log^2 n \log m)$ for a decade [10]. Combined with the result of [14] on the inapproximability of GST, this also shows that MGL, and hence MLUFL with metric connection costs, cannot be approximated to better than a $\Omega(\log m)$-factor unless $NP \subseteq ZTIME\,(n^{\mathrm{polylog}(n)})$.

Given the above hardness result, we investigate certain well-motivated special cases of MLUFL and obtain significantly improved performance guarantees. In Section 2.2, we consider the case where the connection costs form a metric, which is a scalar multiple of the d-metric (i.e., $d_{uv} = c_{uv}/M$, where $M \geq 1$; the problem is trivial if $M < 1$). For ex-

ample, in a supply-chain logistics problem, this models a natural setting where the connection of clients to facilities, and the activation of facilities both proceed along the same transportation network at different speeds. We obtain a *constant-factor* approximation algorithm for this problem. In Section 2.3, we consider the *uniform* MLUFL problem, which is the special case where the time-metric is uniform. Uniform MLUFL already generalizes MSSC (and also UFL). For uniform MLUFL with metric connection costs (i.e., metric uniform MLUFL), we devise a 10.78-approximation algorithm. (Without metricity, the problem becomes set-cover hard, and we obtain a simple matching $O(\log m)$-approximation.) The chief novelty here lies in the technique used to obtain this result. We give a simple generic reduction (Theorem 4) that shows how to reduce the metric uniform MLUFL problem with facility costs to one *without* facility costs, in conjunction with an algorithm for UFL. This reduction is surprisingly robust and versatile and yields, for example, $O(1)$-approximations for *metric uniform k-median* (i.e., metric uniform MLUFL where at most k facilities may be opened), and MLUFL with non-uniform latency costs.

We obtain our approximation bounds by rounding the optimal solution to a suitable linear-programming (LP) relaxation of the problem. In Section 3, we leverage this to obtain some interesting insights about the special case of ML, which we believe cast new light on the problem since all previous approximation algorithms for ML are based on combinatorial approaches. In particular, we present an LP-relaxation for ML, and prove that the integrality gap of these relaxations is *upper bounded* by a (small) constant. Our LP is a specialization of our LP-relaxation for MLUFL. Interestingly, the integrality-gap bound for this LP relies only on the fact that the natural LP relaxation for TSP has constant integrality gap. In contrast, the various known algorithms for ML [2, 6, 1] all utilize algorithms for the arguably harder k-MST problem or its variants. In the full version [4], we describe a second LP relaxation with exponentially-many variables, one for every path (or tree) of a given length bound, where the separation oracle for the dual problem is a rooted path (or tree) orienteering problem: given rewards on the nodes and metric edge costs, find a (simple) path rooted at r of length at most B that gathers maximum reward. We prove that even a bicriteria approximation for the orienteering problem yields an approximation for ML while losing a constant factor. This connection between orienteering and ML is known [7]. But we feel that our alternate proof, where the orienteering problem appears as the separation oracle required to solve the dual LP, offers a more illuminating explanation of the relation between the approximability of the two problems. Our LP-

rounding algorithms to prove the constant integrality gaps exploit various ideas developed for scheduling (e.g., α-points) and polyhedral insights for TSP. This suggests that the wealth of LP based machinery could be leveraged for ML as well; we suspect that our LP-relaxations are in fact much better than what we have accounted for.

LP-based techniques tend to be fairly versatile and can be adapted to handle more general variants of the problem. Our algorithms and analyses extend with little effort to handle various generalizations of MLUFL (and hence, ML). One such example (see Section 4) is the setting where the latency-cost of a client is a function (of bounded growth) of the time taken to reach the facility serving it. This yields an approximation algorithm for the $\mathcal{L}_p$-norm generalization of MLUFL, where we take the $\mathcal{L}_p$-norm of the client latencies (instead of the $\mathcal{L}_1$-norm) in the objective function; these norms tradeoff efficiency with fairness making them an appealing measure to consider. Another notable extension is the k-route version, where we may use k paths starting at r to traverse the open facilities.

Related work. There is a vast amount of literature on facility location and vehicle routing; we refer the reader to [4] for a more-detailed discussion of related work. The work that is most closely related to ours is due to Gupta et al. [13], who independently and concurrently also proposed the minimum group latency (MGL) problem (which they arrive at in the course of solving a different problem), and obtain results similar to ours for MGL. They also obtain an $O(\log^2 n)$-approximation for MGL, and a hardness of approximation for MGL via the reduction from GST to MGL with an $O(\log m)$-factor loss (see also [16]). They reduce MGL to a series of "group orienteering" problems, which they solve using a subroutine due to Charikar et al. [5]. It is not clear how their combinatorial techniques can be extended to handle facility-opening costs in MLUFL.

2 LP-rounding approximation algorithms for MLUFL

We obtain a linear program for MLUFL as follows. We may assume that $d_{ii'}$ is integral for all $i, i' \in \mathcal{F} \cup \{r\}$. Let E denote the edge-set of the complete graph on $\mathcal{F} \cup \{r\}$ and let $d_{max} := \max_{e \in E} d_e$. Let $\mathsf{T} \leq \min\{n, m\} d_{\max}$ be a known upper bound on the maximum activation time of an open facility in an optimal solution. For every facility i, client j, and time $t \leq \mathsf{T}$, we have a variable $y_{i,t}$ indicating if facility i is opened at time t or not, and a variable $x_{ij,t}$ indicating whether client j connects to facility i at time t. Also, for every edge $e \in E$ and time t, we introduce a variable $z_{e,t}$ which denotes if edge e has been traversed *by* time t. Throughout, we use i to

index the facilities in $\mathcal{F}$, j to index the clients in $\mathcal{D}$, t to index the time units in $[\mathsf{T}] := \{1, \ldots, \mathsf{T}\}$, and e to index the edges in E.

$$\min \quad \sum_{i,t} f_i y_{i,t} + \sum_{j,i,t} (c_{ij} + t) x_{ij,t} \tag{P}$$

$$\text{s.t.} \quad \sum_{i,t} x_{ij,t} \geq 1 \qquad \forall j; \qquad x_{ij,t} \leq y_{i,t} \quad \forall i, j, t$$

$$\sum_{e} d_e z_{e,t} \leq t \qquad \forall t \tag{2}$$

$$\sum_{e \in \delta(S)} z_{e,t} \geq \sum_{i \in S, t' \leq t} x_{ij,t'} \qquad \forall t, S \subseteq \mathcal{F}, j \tag{3}$$

$$x_{ij,t}, y_{i,t}, z_{e,t} \geq 0 \qquad \forall i, j, t, e; \qquad y_{i,t} = 0 \qquad \forall i, t \text{ with } d_{ir} > t.$$

The first two constraints encode that each client is connected to some facility at some time, and that if a client is connected to a facility i at time t, then i must be open at time t. Constraint (2) ensures that at most t "distance" is covered by the tour on facilities by time t, and (3) ensures that if a client is connected to i by time t, then the tour must have visited i by time t. We assume here that $\mathsf{T} = \mathsf{poly}(m)$; this assumption can be removed with a loss of an $(1+\varepsilon)$ factor (see [4]). Thus, (P) can be solved efficiently since one can efficiently separate over the constraints (3). Let (x, y, z) be an optimal solution to (P), and OPT denote its objective value. For a client j, define $C_j^* = \sum_{i,t} c_{ij} x_{ij,t}$, and $L_j^* = \sum_{i,t} t x_{ij,t}$. We devise various approximation algorithms for MLUFL by rounding (x, y, z).

2.1 An $O\big(\log n \cdot \max\{\log n, \log m\}\big)$-approximation algorithm

We give an overview of the algorithm. Let $N_j = \{i \in \mathcal{F} : c_{ij} \leq 4C_j^*\}$ be the set of facilities "close" to j, and define τ_j as the earliest time t such that $\sum_{i \in N_j, t' \leq t} x_{ij,t'} \geq \frac{2}{3}$. By Markov's inequality, we have $\sum_{i \in N_j} \sum_t x_{ij,t} \geq \frac{3}{4}$ and $\tau_j \leq 12 L_j^*$. It is easiest to describe the algorithm assuming first that the time-metric d is a tree metric. Our algorithm runs in phases, with phase ℓ corresponding to time $t_\ell = 2^\ell$. In each phase, we compute a random subtree rooted at r of "low" cost such that for every client j with $\tau_j \leq t_\ell$, with constant probability, this tree contains a facility in N_j. To compute this tree, we utilize the rounding procedure of Garg-Konjevod-Ravi (GKR) for the *group Steiner tree* (GST) problem [10] (see Theorem 1 below), by creating a group for each client j with $\tau_j \leq t_\ell$ comprising of, roughly speaking, the facilities in N_j. We open all the facilities included in the subtree, and obtain a tour via the standard trick of doubling all edges and performing an Eulerian tour with possible shortcutting. The overall tour is a concatenation of all the tours obtained in the various

phases. For each client j, we consider the first tree that contains a facility from N_j (which must therefore be open), and connect j to such a facility.

Given the result for tree metrics, an oft-used idea to handle the case when d is not a tree metric is to approximate it by a distribution of tree metrics with $O(\log n)$ distortion [8]. Our use of this idea is however slightly subtle Instead of moving to a distribution over tree metrics up front, in each phase ℓ, we use the results of [5, 8] to *deterministically* obtain a tree $\mathcal{T}_\ell$ with edge weights $\{d_{\mathcal{T}_\ell}(e)\}$, such that the resulting tree metric dominates d and $\sum_{e=(i,i')} d_{\mathcal{T}_\ell}(i,i') z_{e,t_\ell} = O(\log n) \sum_e d_e z_{e,t_\ell}$. This deterministic choice allows to extend our algorithm and analysis effortlessly to the setting where the latency-cost in the objective function is measured by a more general function of the client-latencies. Algorithm 1 is a detailed description of the algorithm. Let $\tau_{\max} = \max_j \tau_j$.

Theorem 1 ([5, 8]). *Given any edge weights $\{\mathfrak{z}_e\}_{e\in E}$, one can deterministically construct a weighted tree $\mathcal{T}$ having leaf-set $\mathcal{F} \cup \{r\}$, leading to a tree metric, $d_{\mathcal{T}}(\cdot)$, such that, for any $i, i' \in \mathcal{F} \cup \{r\}$, we have: (i) $d_{\mathcal{T}}(i,i') \geq d_{ii'}$, and (ii) $\sum_{e=(i,i')\in E} d_{\mathcal{T}}(i,i')\mathfrak{z}_{i,i'} = O(\log n) \sum_e d_e \mathfrak{z}_e$.*

Theorem 2 ([10]). *Consider a tree $\mathcal{T}$ rooted at r with n leaves, subsets $G_1, \ldots, G_p$ of leaves, and fractional values $\mathfrak{z}_e$ on the edges of $\mathcal{T}$ satisfying $\mathfrak{z}(\delta(S)) \geq \nu_j$ for every group G_j and node-set S such that $G_j \subseteq S$, where $\nu_j \in \left[\frac{1}{2}, 1\right]$. There exists a randomized polytime algorithm, henceforth called the GKR algorithm, that returns a rooted subtree $T'' \subseteq \mathcal{T}$ such that (i) $\Pr[e \in T''] \leq \mathfrak{z}_e$ for every edge $e \in \mathcal{T}$; and (ii) $\Pr[T'' \cap G_j = \emptyset] \leq \exp\left(-\frac{\nu_j}{64 \log_2 n}\right)$ for every group G_j.*

Analysis. Consider any phase ℓ. For any subset S of nodes of the corresponding tree $\mathcal{T}'_\ell$ with $r \notin S$, and any $N'_j \subseteq S$ where $j \in D_\ell$, we have $\mathfrak{z}(\delta_{\mathcal{T}'_\ell}(S)) \geq \sum_{i\in N_j, t\leq t_\ell} x_{ij,t} \geq 2/3$. This follows from the constraint (3) in the LP. Using Theorem 2, we get the following lemma which bounds the probability of failure in step A1.3.

Lemma 1. *In any phase ℓ, with probability $1 - \frac{1}{\mathsf{poly}(m)}$, we obtain the desired tree T'_ℓ in step A1.3. Also, $\Pr[T'_\ell \cap N'_j \neq \emptyset] \geq 5/9$ for all $j \in D_\ell$.*

Since each client j is connected to a facility in N_j, the total connection cost is at most $4 \sum_j C^*_j$. Furthermore, from Lemma 1 we get that for every client $j \in D_\ell$, the probability that a facility in N_j is included in the tree T'_ℓ, and hence opened in phase ℓ, is at least $\frac{5}{9}$. The facility-cost incurred in a phase is $O(\log n) \sum_{i,t} f_i y_{i,t}$, and since $\tau_{\max} \leq \mathsf{T} = \mathsf{poly}(m)$, the number of phases is $O(\log m)$, so this bounds the facility-opening cost incurred. Also,

Algorithm 1 Given: a fractional solution (x, y, z) to (P).

A1. In each phase $\ell = 0, 1, \ldots, \mathcal{N} := \lceil \log_2(2\tau_{\max}) + 4\log_2 m \rceil$, we do the following. Let $t_\ell = \min\{2^\ell, \mathsf{T}\}$.

A1.1. Use Theorem 1 with edge weights $\{z_{e,t_\ell}\}$ to obtain a tree $\mathcal{T}_\ell = \big(V(\mathcal{T}_\ell), E(\mathcal{T}_\ell)\big)$. Extend $\mathcal{T}_\ell$ to a tree $\mathcal{T}'_\ell$ by adding a dummy leaf edge (i, v_i) of cost f_i to $\mathcal{T}_\ell$ for each facility i. Let $E' = \{(i, v_i) : i \in \mathcal{F}\}$.

A1.2. Map the LP-assignment $\{z_{e,t_\ell}\}_{e\in E}$ to an assignment $\mathfrak{z}$ on the edges of $\mathcal{T}'_\ell$ by setting $\mathfrak{z}_e = \sum_{e \text{ lies on the unique } i\text{-}i' \text{ path in } \mathcal{T}_\ell} z_{ii',t_\ell}$ for all $e \in E(\mathcal{T}_\ell)$, and $\mathfrak{z}_e = \sum_{t\le t_\ell} y_{i,t}$ for all $e = (i, v_i) \in E'$.

A1.3. Define $D_\ell = \{j : \tau_j \le t_\ell\}$. For each client $j \in D_\ell$, we define the group $N'_j = \{v_i : i \in N_j\}$. We now compute a subtree T'_ℓ of $\mathcal{T}'_\ell$ as follows. We obtain $N := \log_2 m$ subtrees $T''_1, \ldots, T''_N$. Each tree T''_r is obtained by executing the GKR algorithm $192\log_2 n$ times on the tree $\mathcal{T}'_\ell$ with groups $\{N'_j\}_{j\in D_\ell}$, and taking the union of all the subtrees returned. Note that we may assume that $i \in T''_r$ iff $(i, v_i) \in T''_r$. Set T'_ℓ to be the first tree in $\{T''_1, \ldots, T''_N\}$ satisfying (i) $\sum_{(i,v_i)\in E(T'_\ell)} f_i \le 40 \cdot 192\log_2 n \sum_{(i,v_i)\in E'} f_i \mathfrak{z}_{i,v_i}$ and (ii) $\sum_{e\in E(T'_\ell)\setminus E'} d_{\mathcal{T}_\ell}(e) \le 40 \cdot 192\log_2 n \sum_{e\in E(\mathcal{T}_\ell)} d_{\mathcal{T}_\ell}(e)\mathfrak{z}_e$; if no such tree exists, the algorithm fails.

A1.4. Now remove all the dummy edges from T'_ℓ, open all the facilities in the resulting tree, and convert the resulting tree into a tour Tour_ℓ traversing all the opened facilities. For every unconnected client j, we connect j to a facility in N_j if some such facility is open (and hence part of Tour_ℓ).

A2. Return the concatenation of the tours Tour_ℓ for $\ell = 0, 1, \ldots, \mathcal{N}$ shortcutting whenever possible. This induces an ordering of the open facilities. If some client is left unconnected, we say that the algorithm has failed.

since the probability that j is not connected (to a facility in N_j) in phase ℓ decreases geometrically (at a rate less than $1/2$) with ℓ when $t_\ell \ge \tau_j$, one can argue that (a) with very high probability (i.e., $1 - 1/\mathsf{poly}(m)$), each client j is connected to some facility in N_j, and (b) the expected latency-cost of j is at most $O(\log n)\sum_{e\in E(\mathcal{T}_\ell)} d_{\mathcal{T}_\ell}(e)\mathfrak{z}_e = O(\log^2 n)\tau_j$.

Lemma 2. *The probability that a client j is not connected by the algorithm is at most $1/m^4$. Let L_j be the random variable equal to j's latency-cost if the algorithm succeeds and 0 otherwise. Then $\mathrm{E}\big[L_j\big] = O(\log^2 n)t_{\ell_j}$, where ℓ_j $(= \lceil \log_2 \tau_j \rceil)$ is the smallest ℓ such that $t_\ell \ge \tau_j$.*

Proof. Let P_j be the random phase in which j gets connected; let $P_j := \mathcal{N}+1$ if j remains unconnected. We have $\Pr[P_j \ge \ell] \le \big(\frac{4}{9}\big)^{(\ell-\ell_j)}$ for $\ell \ge \ell_j$. The algorithm proceeds for at least $4\log_2 m$ phases after phase ℓ_j, so $\Pr[j$ is not connected after $\mathcal{N}$ phases$] \le 1/m^4$. Now, $L_j \le \sum_{\ell\le P_j} d(\mathsf{Tour}_\ell) \le 2\sum_{\ell\le P_j}\sum_{e\in E(T'_\ell)\setminus E'} d_{\mathcal{T}_\ell}(e)$. The RHS is $O(\log n)\sum_{\ell\le P_j}\sum_{e\in E(\mathcal{T}_\ell)} d_{\mathcal{T}_\ell}(e)\mathfrak{z}_e = O(\log^2 n)\sum_{\ell\le P_j} t_\ell$ from step A1.3. So $\mathrm{E}\big[L_j\big] = O(\log^2 n)\sum_{\ell=0}^{\mathcal{N}} \Pr[P_j \ge \ell] \cdot t_\ell \le O(\log^2 n)\big[\sum_{\ell=0}^{\ell_j} t_\ell + \sum_{\ell>\ell_j} t_\ell \cdot \big(\frac{4}{9}\big)^{(\ell-\ell_j)}\big] = O(\log^2 n)t_{\ell_j}$.

Theorem 3. *Algorithm 1 succeeds with probability* $1 - 1/\operatorname{poly}(m)$*, and returns a solution of expected cost* $O\big(\log n \cdot \max\{\log n, \log m\}\big) \cdot OPT$*.*

2.2 MLUFL with related metrics

Here, we consider the MLUFL problem when the facilities, clients, and the root r are located in a common metric space that defines the connection-cost metric (on $\mathcal{F} \cup \mathcal{D} \cup \{r\}$), and we have $d_{uv} = c_{uv}/M$ for all $u, v \in \mathcal{F} \cup \mathcal{D} \cup \{r\}$. We call this problem, *related MLUFL*, and design an $O(1)$-approximation algorithm for it.

The algorithm follows a similar outline as Algorithm 1. As before, we build the tour on the open facilities by concatenating tours obtained by "Eulerifying" GST's rooted at r of geometrically increasing length. At a high level, the improvement in the approximation arises because one can now obtain these trees without resorting to Theorem 2 and losing $O(\log n)$-factors in process. Instead, since the d- and c- metrics are related, we obtain a GST on the relevant groups by using a Steiner tree algorithm.

As before, N_j denotes the facilities "close by" (in the c-metric) client j and $\tau_j = O(L_j^*)$. In each phase ℓ we want a GST for the groups N_j for which $\tau_j \le t_\ell$. To obtain this, we first do a facility-location-style clustering of the clients (with $\tau_j \le t_\ell$) to obtain some cluster centers whose N_js are disjoint. We contract these disjoint N_js (of cluster centers) to supernodes and find a minimum Steiner tree connecting these. Since facilities in N_j are close by in the c-metric, and *since the d-metric and c-metric are related*, they are close by in the d-metric as well. Thus, the supernodes in the Steiner tree can be "opened up" to give the GST of not too large cost.

Deciding which facilities to open is tricky since we cannot open facilities in each phase. This is because although N_js are disjoint in an individual phase, they might overlap with N_ks from a different phase. To overcome this, we consider the collection $\mathcal{C}$ of cluster centers created in all the phases, and pick a maximal subset $\mathcal{C}' \subseteq \mathcal{C}$ that yields disjoint N_j's by greedily considering clusters in increasing C_j^* order. We open the cheapest facility i in each of these N_j's, this bounds the facility cost. However, there could be a client $k \in \mathcal{C} \setminus \mathcal{C}'$ which got removed from $\mathcal{C}$ since N_k overlapped with N_j; this k must be connected to i. The issue is that τ_k could be much smaller than τ_j, and thus i needs to be connected to the tree $\mathcal{T}_\ell$ where ℓ is the phase when N_k got connected. To argue this doesn't affect the latency cost too much we once again use the relation between the d-metric and c-metric to show that the total increase in latency cost is at most a constant fraction more.

2.3 MLUFL with a uniform time-metric

We now consider the special case of MLUFL, referred to as *uniform MLUFL*, where the time-metric d is uniform, that is, $d_{ii'} = 1$ for all $i, i' \in \mathcal{F} \cup \{r\}$. When the connection costs form a metric, we call it the *metric uniform MLUFL*. We consider the following simpler LP-relaxation of the problem, where the time t now ranges from 1 to n.

$$\min \quad \sum_{i,t} f_i y_{i,t} + \sum_{j,i,t} (c_{ij} + t) x_{ij,t} \qquad \text{subject to} \qquad \text{(Unif-P)}$$

$$\sum_{i,t} x_{ij,t} \geq 1 \ \forall j; \quad x_{ij,t} \leq y_{i,t} \ \forall i,j,t; \quad \sum_{i} y_{i,t} \leq 1 \ \forall t; \quad x_{ij,t}, y_{i,t} \geq 0 \ \forall i,j,t.$$

The main result of this section is Theorem 4, which shows that a ρ_{UFL}-approximation algorithm for UFL and a γ-approximation algorithm for uniform ZFC MLUFL (uniform MLUFL with zero facility costs) can be combined to yield a $(\rho_{\mathsf{UFL}} + 2\gamma)$-approximation algorithm for metric uniform MLUFL. One can show that $\gamma \leq 9$ (ZFC MLUFL can be reduced to MSSC incurring a constant-factor loss; see [4]), and $\rho_{\mathsf{MLUFL}} \leq 1.5$ [3]; this gives a 19.5 approximation. In the full version, we show that the analysis can be refined to yield an improved 10.773 approximation.

Theorem 4. *Given a ρ_{UFL}-approximation algorithm $\mathcal{A}_1$ for UFL, and a γ-approximation algorithm $\mathcal{A}_2$ for uniform ZFC MLUFL, one can obtain a $(\rho_{\mathsf{UFL}} + 2\gamma)$-approximation algorithm for metric uniform MLUFL.*

Proof. Let $\mathcal{I}$ denote the metric uniform MLUFL instance, and O^* denote the cost of an optimal integer solution. Let $\mathcal{I}_{\mathsf{UFL}}$ be the UFL instance obtained form $\mathcal{I}$ by ignoring the latency costs, and $\mathcal{I}_{\mathsf{ZFC}}$ be the ZFC MLUFL instance obtained from $\mathcal{I}$ by setting all facility costs to zero. Let O^*_{UFL} and O^*_{ZFC} denote respectively the cost of the optimal (integer) solutions to these two instances. Clearly, we have $O^*_{\mathsf{UFL}}, O^*_{\mathsf{ZFC}} \leq O^*$. We use $\mathcal{A}_1$ to obtain a near-optimal solution to $\mathcal{I}_{\mathsf{UFL}}$: let F_1 be the set of facilities opened and let $\sigma_1(j)$ denote the facility in F_1 to which client j is assigned. So we have $\sum_{i \in F_1} f_i + \sum_j c_{\sigma_1(j)j} \leq \rho_{\mathsf{UFL}} \cdot O^*_{\mathsf{UFL}}$. We use $\mathcal{A}_2$ to obtain a near-optimal solution to $\mathcal{I}_{\mathsf{ZFC}}$: let F_2 be the set of open facilities, $\sigma_2(j)$ be the facility to which client j is assigned, and $\pi(i)$ be the position of facility i. So we have $\sum_j \big(c_{\sigma_2(j)j} + \pi(\sigma_2(j))\big) \leq \gamma \cdot O^*_{\mathsf{ZFC}}$.

We now combine these solutions as follows. For each facility $i \in F_2$, let $\mu(i) \in F_1$ denote the facility in F_1 that is nearest to i. We open the set $F = \{\mu(i) : i \in F_2\}$ of facilities. The position of facility $i \in F$ is set to $\min_{i' \in F_2 : \pi(i') = i} \pi(i')$. Each facility in F is assigned a distinct position this

way, but some positions may be vacant. Clearly we can always convert the above into a proper ordering of F where each facility $i \in F$ occurs at position $\kappa(i) \leq \min_{i' \in F_2 : \pi(i') = i} \pi(i')$. Finally, we assign each client j to the facility $\phi(j) = \mu(\sigma_2(j)) \in F$. Note that $\kappa(\phi(j)) \leq \pi(\sigma_2(j))$ (by definition). For a client j, we now have $c_{\phi(j)j} \leq c_{\sigma_2(j)\mu(\sigma_2(j))} + c_{\sigma_2(j)j} \leq c_{\sigma_2(j)\sigma_1(j)} + c_{\sigma_2(j)j} \leq c_{\sigma_1(j)j} + 2c_{\sigma_2(j)j}$. Thus, the total cost of the resulting solution is at most $\sum_{i \in F_1} f_i + \sum_j \big(c_{\sigma_1(j)j} + 2c_{\sigma_2(j)j} + \pi(\sigma_2(j))\big) \leq (\rho_{\mathsf{UFL}} + 2\gamma) \cdot O^*$.

3 LP-relaxations and algorithms for ML

In this section, we give an LP-relaxation for the ML problem and prove that it has a constant integrality gap. In the full version, we describe another LP-relaxation for ML for which also we prove a constant upper bound on the integrality gap. We believe that our LP-relaxations are stronger than what we have accounted for, and conjecture that the integrality gap of the second LP is at most 3.59, the current best known approximation factor for ML. The second LP also gives an illuminating explanation of the relation between ML and orienteering.

Let $G = (\mathcal{D} \cup \{r\}, E)$ be the complete graph on $N = |\mathcal{D}| + 1$ nodes with edge weights $\{d_e\}$ that form a metric. Let r be the root node at which the path visiting the nodes must originate. We use e to index E and j to index the nodes. We have variables $x_{j,t}$ for $t \geq d_{jr}$ to denote if j is visited at time t, and $z_{e,t}$ to denote (as before) if e has been traversed by time t (where t ranges from 1 to T); for convenience, we think of $x_{j,t}$ as being defined for all t, with $x_{j,t} = 0$ if $d_{j,r} > t$. (As before, one can move to a polynomial-size LP losing a $(1+\epsilon)$-factor.)

$$\min \sum_{j,t} t x_{j,t} \quad \text{subject to} \tag{LP1}$$

$$\sum_t x_{j,t} \geq 1 \ \forall j; \quad \sum_e d_e z_{e,t} \leq t \ \forall t; \quad \sum_{e \in \delta(S)} z_{e,t} \geq \sum_{t' \leq t} x_{j,t'} \ \forall t, S \subseteq \mathcal{D}, j \in S; \quad x, z \geq 0.$$

Theorem 5. *The integrality gap of* (LP1) *is at most* 10.78.

Proof. Let (x, z) be an optimal solution to (LP1), and $L_j^* = \sum_t t x_{j,t}$. For $\alpha \in [0, 1]$, define the *α-point of j*, $\tau_j(\alpha)$, to be the smallest t such that $\sum_{t' \leq t} x_{jt'} \geq \alpha$. Let $D_t(\alpha) = \{j : \tau_j(\alpha) \leq t\}$. We round (x, z) as follows. We pick $\alpha \in (0, 1]$ according to the density function $q(x) = 2x$. For each time t, using the parsimonious property (see [11]), one can see that $(2z/\alpha)$ is a feasible solution to the sub-tour elimination LP for TSP on the vertices $r \cup D_t(\alpha)$. Then we utilize the $\frac{3}{2}$-integrality-gap of this LP [20, 18], to round $\frac{2z}{\alpha}$ and obtain a tour on $\{r\} \cup D_t(\alpha)$ of cost $C_t(\alpha) \leq \frac{3}{\alpha} \cdot \sum_e d_e z_{e,t} \leq \frac{3t}{\alpha}$. We now use Lemma 3 to combine these tours.

Lemma 3 ([12] paraphrased). *Let* $\mathsf{Tour}_1, \ldots, \mathsf{Tour}_k$ *be tours containing* r*, with* Tour_i *having cost* C_i *and containing* N_i *nodes, where* $N_0 := 1 \leq N_1 \leq \ldots \leq N_k = N$*. One can find tours* $\mathsf{Tour}_{i_1}, \ldots, \mathsf{Tour}_{i_b=k}$*, and concatenate them suitably to obtain latency at most* $\frac{3.59}{2} \sum_i C_i(N_i - N_{i-1})$*.*

The tours we obtain for the different times are nested (as the $D_t(\alpha)$s are nested). So $\sum_{t\geq 1} C_t(\alpha)(|D_t(\alpha)| - |D_{t-1}(\alpha)|) = \sum_{j,t:j\in D_t(\alpha)\setminus D_{t-1}(\alpha)} C_t(\alpha) = \sum_j C_{\tau_j(\alpha)}(\alpha) \leq 3\sum_j \frac{\tau_j(\alpha)}{\alpha}$. Using Lemma 3, and taking expectation over α (note that $\mathrm{E}\big[\frac{\tau_j(\alpha)}{\alpha}\big] \leq 2L_j^*$), we get total latency cost at most $10.78\sum_j L_j^*$.

Interestingly, note that in the above proof we did not need any procedure to solve k-MST or its variants, which all previously known algorithms for ML use as a subroutine. Rather, we just needed the integrality gap of the subtour-elimination LP to be a constant.

4 Extensions

Latency cost functions. Consider the setting where the latency-cost of client j is given by λ(time taken to reach the facility serving j), where $\lambda(.)$ is a non-decreasing function; the goal, as before, is to minimize the sum of the facility-opening, client-connection, and client-latency costs. Say that λ has growth at most p if $\lambda(cx) \leq c^p\lambda(x)$ for all $x \geq 0,\ c \geq 1$. It is not hard to see that for concave λ, we obtain the same performance guarantees as those obtained in Section 2. For convex λ, we obtain an $O\big(\max\{(p\log^2 n)^p, p\log n\log m\}\big)$-approximation algorithm for convex latency functions of growth p. As a *corollary*, we obtain an approximation guarantee for $\mathcal{L}_p$-MLUFL, where we seek to minimize the facility-opening cost + client-connection cost + the $\mathcal{L}_p$-norm of client-latencies.

Theorem 6. *There is an* $O\big(\max\{(p\log^2 n)^p, p\log n\log m\}\big)$*-approximation algorithm for* MLUFL *with convex monotonic latency functions of growth* p*. This yields an* $O\big(p\log n\max\{\log n, \log m\}\big)$ *approximation for* $\mathcal{L}_p$*-*MLUFL*.*

In *k-route length-bounded* MLUFL, we are given a budget B and we may use (at most) k paths starting at r of (d-) length at most B to traverse the open facilities and activate them. (So with $B = \infty$, this generalizes the *k-traveling repairmen* problem [7].) Our algorithms easily extend to give: (a) a bicriteria $\big(\text{polylog}, O(\log^2 n)\big)$-approximation for the general k-route MLUFL problem where we violate the budget by a $O(\log^2 n)$ factor; (b) an $(O(1), O(1))$ approximation for MLUFL and ML with related metrics; and (c) a (*unicriterion*) $O(1)$ approximation for metric uniform MLUFL. These guarantees extend to latency functions of bounded growth. In particular, we obtain an $O(1)$ *approximation* for the $\mathcal{L}_p$-norm k-traveling repairmen problem; this is the *first* approximation guarantee for this problem.

References

1. A. Archer, A. Levin, and D. Williamson. A faster, better approximation algorithm for the minimum latency problem. *SIAM J. Comput.*, 37(5):1472–1498, 2008.
2. A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan. The Minimum Latency Problem. *Proc. 26th STOC*, pages 163–171, 1994.
3. J. Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. In *Proc. 10th APPROX*, pages 29–43, 2007.
4. D. Chakrabarty, and C. Swamy. Facility Location with Client Latencies: Linear Programming based Techniques for Minimum Latency Problems. *http://arxiv.org/abs/1009.2452*
5. M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: deterministic approximation algorithms for Group Steiner Trees and k-median. In *Proceedings of the 30th STOC*, pages 114–123, 1998.
6. K. Chaudhuri, P. B. Godfrey, S. Rao, and K. Talwar. Paths, Trees and Minimum Latency Tours. In *Proceedings of 44th FOCS*, pages 36–45, 2003.
7. J. Fakcharoenphol, C. Harrelson, and S. Rao The k-traveling repairman problem. *ACM Trans. on Alg.*, Vol 3, Issue 4, Article 40, 2007.
8. J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proc. 35th STOC*, pages 448–455, 2003.
9. U. Feige, L. Lovász, and P. Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004.
10. N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000.
11. M. Goemans and D. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Math. Programming*, 60:145–166, 1993.
12. M. Goemans and J. Kleinberg. An improved approximation ratio for the minimum latency problem. In *Proceedings of 7th SODA*, pages 152–158, 1996.
13. A. Gupta, R. Krishnaswamy, V. Nagarajan, and R. Ravi. Approximation Algorithms for Optimal Decision Trees and Adaptive TSP Problems. In *Proceedings of 37th ICALP*, pages 690–701.
14. E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of 35th STOC*, pages 585–594, 2003.
15. P. Mirchandani and R. Francis, eds. *Discrete Location Theory*. John Wiley and Sons, Inc., New York, 1990.
16. V. Nagarajan. *Approximation Algorithms for Sequencing Problems*. Ph.D. thesis, Tepper School of Business, Carnegie Mellon University, 2009.
17. D. B. Shmoys, É. Tardos, and K. I. Aardal. Approximation algorithms for facility location problems. In *Proceedings of 29th STOC*, pages 265–274, 1997.
18. D. Shmoys and D. Williamson. Analyzing the Held-Karp TSP bound: a monotonicity property with application. *Inf. Process. Lett.*, 35(6):281–285, 1990.
19. P. Toth and D. Vigo, eds. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 2002.
20. L. Wolsey. Heuristic analysis, linear programming and branch and bound. *Mathematical Programming Study*, 13:121-134, 1980.