Efficient Online Clustering with Moving Costs

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work we consider an online learning problem, called *Online k-Clustering with Moving Costs*, at which a *learner* maintains a set of k facilities over T rounds so as to minimize the connection cost of an adversarially selected sequence of clients. The *learner* is informed on the positions of the clients at each round t only after its facility-selection and can use this information to update its decision in the next round. However, updating the facility positions comes with an additional moving cost based on the moving distance of the facilities. We present the first $\mathcal{O}(\log n)$-regret polynomial-time online learning algorithm guaranteeing that the overall cost (connection + moving) is at most $\mathcal{O}(\log n)$ times the time-averaged connection cost of the *best fixed solution*. Our work improves on the recent result of Fotakis et al. [30] establishing $\mathcal{O}(k)$-regret guarantees *only* on the connection cost.

1 Introduction

Due to their various applications in diverse fields (e.g. machine learning, operational research, data science etc.), *clustering problems* have been extensively studied. In the well-studied k-median problem, given a set of clients, k facilities should be placed on a metric with the objective to minimize the sum of the distance of each client from its closest center [54, 14, 13, 66, 6, 43, 51, 64, 50, 15, 53, 3].

In many modern applications (e.g., epidemiology, social media, conference, etc.) the positions of the clients are not *static* but rather *evolve over time* [56, 55, 63, 58, 23, 5]. For example the geographic distribution of the clients of an online store or the distribution of Covid-19 cases may drastically change from year to year or respectively from day to day [30]. In such settings it is desirable to update/change the positions of the facilities (e.g., compositions of warehouses or Covid test-units) so as to better serve the time-evolving trajectory of the clients.

The clients' positions may change in complex and unpredictable ways and thus an *a priori knowledge* on their trajectory is not always available. Motivated by this, a recent line of research studies clustering problems under the *online learning framework* by assuming that the sequence of clients' positions is *unknown* and *adversarially selected* [18, 28, 16, 30]. More precisely, a *learner* must place k facilities at each round $t \geq 1$ without knowing the positions of clients at round t which are revealed to the learner only after its facility-selection. The learner can use this information to update its decision in the next round; however, moving a facility comes with an additional moving cost that should be taken into account in the learner's updating decision, e.g. moving Covid-19 test-units comes with a cost [18, 28].

Building on this line of works, we consider the following online learning problem:

Problem 1 (*Online k-Clustering with Moving Costs*)**.** *Let $G(V, E, w)$ be a weighted graph with $|V| = n$ vertices and k facilities. At each round $t = 1, \ldots, T$:*

1. *The learner selects $F_t \subseteq V$, with $|F_t| = k$, at which facilities are placed.*
2. *The adversary selects the clients' positions, $R_t \subseteq V$.*

3. The learner learns the clients' positions R_t and suffers

$$\textit{cost} = \sum_{j \in R_t} \underbrace{\min_{i \in F_t} d_G(j,i)}_{\textit{connection cost of client } j} + \underbrace{\gamma \cdot M_G(F_{t-1}, F_t)}_{\textit{moving cost of facilities}}$$

where $d_G(j,i)$ is the distance between vertices $i, j \in V$; $M_G(F_{t-1}, F_t)$ is the minimum overall distance required to move k facilities from F_{t-1} to F_t; and $\gamma \geq 0$ is the facility-weight.

An *online learning algorithm* for Problem 1 tries to minimize the overall (connection + moving) cost by placing k facilities at each round $t \geq 1$ based only on the previous positions of clients $R_1, \ldots, R_{t-1}$. To the best of our knowledge, Problem 1 was first introduced in [18][1]. If for any sequence of clients, the overall cost of the algorithm is at most α times the overall connection cost of the *optimal fixed placement of facilities* F^* then the algorithm is called α-regret, while in the special case of $\alpha = 1$ the algorithm is additionally called *no-regret*.

Problem 1 comes as a special case of the well-studied *Metrical Task System* by considering each of the possible $\binom{n}{k}$ facility placements as a different state. In their seminal work, [11] guarantee that the famous *Multiplicative Weights Update algorithm* (MWU) achieves $(1+\epsilon)$-regret in Problem 1 for any $\epsilon > 0$. Unfortunately, running the MWU algorithm for Problem 1 is not really an option since it requires $\mathcal{O}(n^k)$ time and space complexity. As a result, the following question naturally arises:

Q. *Can we achieve α-regret for Problem 1 with polynomial-time online learning algorithms?*

Answering the above question is a challenging task. Even in the very simple scenario of time-invariant clients, i.e. $R_t = R$ for all $t \geq 1$, an α-regret online learning algorithm must essentially compute an α-*approximate solution* of the k-median problem. Unfortunately the k-median problem cannot be approximated with ratio $\alpha < 1 + 2/e \simeq 1.71$ (unless $\mathrm{NP} \subseteq \mathrm{DTIME}[n^{\log\log n}]$ [42]) which excludes the existence of an $(1+2/e)$-regret polynomial-time online learning algorithm for Problem 1. Despite the fact that many $\mathcal{O}(1)$-approximation algorithms have been proposed for the k-median problem (the best current ratio is $1+\sqrt{3}$ [53]), these algorithms crucially rely on the (offline) knowledge of the whole sequence of clients and most importantly are not designed to handle the moving cost of the facilities [54, 14, 13, 66, 6, 43, 51, 64, 50, 15, 53, 3].

In their recent work, Fotakis et al. [30] propose an $\mathcal{O}(k)$-regret polynomial-time online learning algorithm for Problem 1 *without* moving costs (i.e. the special case of $\gamma = 0$). Their approach is based on designing a *no-regret* polynomial-time algorithm for a *fractional relaxation* of Problem 1 and then using an *online client-oblivious* rounding scheme in order to convert a fractional solution to an integral one. Their analysis is based on the fact that the connection cost of *any possible client* is at most $\mathcal{O}(k)$ times its fractional connection cost. However in order to establish the latter guarantee their rounding scheme performs abrupt changes on the facilities leading to huge moving cost.

Our Contribution and Techniques. In this work, we provide a positive answer to question (**Q**), by designing the first polynomial-time online learning algorithm for Online k-Clustering with Moving Costs that achieves $\mathcal{O}(\log n)$-regret for any $\gamma \geq 0$. The cornerstone idea of our work was to realize that $\mathcal{O}(1)$-regret can be established with a polynomial-time online learning algorithm in the special case of G being a Hierarchical Separation Tree (HST). Then, by using the standard metric embedding result of [25], we can easily convert such an algorithm to an $\mathcal{O}(\log n)$-regret algorithm for general graphs. Our approach for HSTs consists of two main technical steps:

1. We introduce a fractional relaxation of Problem 1 for HSTs. We then consider a specific regularizer on the fractional facility placements, called *Dilated Entropic Regularizer* [26], that takes into account the specific structure of the HST. Our first technical contribution is to establish that the famous *Follow the Leader algorithm* [34] with dilated entropic regularization admits $\mathcal{O}(1)$-regret for any $\gamma \geq 0$.
2. Our second technical contribution is the design of a novel *online client-oblivious* rounding scheme, called $\mathrm{Cut\&Round}$, that converts a fractional solution for HSTs into an integral one. By exploiting the specific HST structure we establish that $\mathrm{Cut\&Round}$, despite not

[1]In [18], an easier version of Problem 1 with 1-*lookahead* is considered, meaning that the learner learns the positions of the clients R_t before selecting F_t. Moreover, G is considered to be the line graph and $\gamma = 1$.

knowing the clients' positions R_t, simultaneously guarantees that (i) the connection cost of each client $j \in R_t$ is upper bounded by its fractional connection cost, and (ii) the expected moving cost of the facilities is at most $\mathcal{O}(1)$ times the fractional moving cost.

Experimental Evaluation. In Section F of the Appendix we experimentally compare our algorithm with the algorithm of Fotakis et al. [30]. Our experiments verify that our algorithm is robust to increases of the facility weight γ while the algorithm of [30] presents a significant cost increase. We additionally experimentally evaluate our algorithm in the MNIST and $\mathrm{CIFAR10}$ datasets. Our experimental evaluations suggest that the $\mathcal{O}(\log n)$-regret bound is a pessimistic upper bound and that in practise our algorithm performs significantly better.

Related Work. As already mentioned, our work most closely relates with the work of Fotakis et al. [30] that provides an $\mathcal{O}(k)$-regret algorithm running in polynomial-time for $\gamma = 0$. [16] also consider Problem 1 for $\gamma = 0$ with the difference that the connection cost of clients is captured through the $k\text{-means}$ objective i.e. the sum of the squared distances. They provide an $(1+\epsilon)$-regret algorithm with $\mathcal{O}\left((k^2/\epsilon^2)^{2k}\right)$ time-complexity that is still exponential in k. [18, 28] study the special case of Problem 1 in which G is the line graph and $\gamma = 1$ while assuming 1-*lookahead* on the request R_t. For $k = 1$, [18] provide an $(1+\epsilon)$-competitive online algorithm meaning that its cost is at most $(1+\epsilon)$ times the cost of the *optimal dynamic solution* and directly implies $(1+\epsilon)$-regret. [28] extended the previous result by providing a 63-competitive algorithm for $k = 2$ on line graphs. Our work also relates with the works of [23] and [4] that study offline approximation algorithms for clustering problems with *time-evolving metrics*. Finally our work is closely related with the research line of online learning in combinatorial domains and other settings of online clustering. Due to space limitations, we resume this discussion in Section A of the Appendix.

2 Preliminaries and Our Results

Let $G(V, E, w)$ be a weighted undirected graph where V denotes the set of vertices and E the set of edges among them. The weight w_e of an edge $e = (i, j) \in E$ denotes the cost of traversing e. Without loss, we assume that $w_e \in \mathbb{N}$ and $w_e \geq 1$ for all edges $e \in E$. The *distance* between vertices $i, j \in V$ is denoted with $d_G(i, j)$ and equals the cost of the minimum cost path from $i \in V$ to $j \in V$. We use $n := |V|$ to denote the cardinality of G and $D_G := \max_{i,j \in V} d_G(i, j)$ to denote its diameter.

Given a placement of facilities $F \subseteq V$, with $|F| = k$, a client placed at vertex $j \in V$ connects to the *closest open facility* $i \in F$. This is formally captured in Definition 1.

Definition 1. *The connection cost of a set of clients $R \subseteq V$ under the facility-placement $F \subseteq V$ with $|F| = k$ equals*

$$C_R(F) := \sum_{j \in R} \min_{i \in F} d_G(j, i)$$

Next, consider any pair of facility-placements $F, F' \subseteq V$ such that $|F| = |F'| = k$. The moving distance between F and F' is the minimum overall distance needed to transfer the k facilities from F to F', formally defined in Definition 2.

Definition 2. *Fix any facility-placements $F, F' \subseteq V$ where $|F| = |F'| = k$. Let Σ be the set of all possible matchings from F to F', i.e. each $\sigma \in \Sigma$ is a one-to-one mapping $\sigma : F \mapsto F'$ with $\sigma(i) \in F'$ denoting the mapping of facility $i \in F$. The moving cost between F and F' equals*

$$M_G(F, F') := \min_{\sigma \in \Sigma} \sum_{i \in F} d_G(i, \sigma(i))$$

At each round $t \geq 1$, an online learning algorithm $\mathcal{A}$ for Problem 1 takes as input all the *previous* positions of the clients $R_1, \ldots, R_{t-1} \subseteq V$ and outputs a facility-placement $F_t := \mathcal{A}(R_1, \ldots, R_{t-1})$ such that $F_t \subseteq V$ and $|F_t| = k$. The performance of an online learning algorithm is measured by the notion of *regret*, which we formally introduce in Definition 3.

Definition 3. *An online learning algorithm $\mathcal{A}$ for Problem 1 is called α-regret with additive regret β if and only if for any sequence of clients $R_1, \ldots, R_T \subseteq V$,*

$$\mathbb{E}\left[\sum_{t=1}^{T} C_{R_t}(F_t) + \gamma \cdot \sum_{t=2}^{T} M_G(F_{t-1}, F_t)\right] \leq \alpha \cdot \min_{|F^*|=k} \sum_{t=1}^{T} C_{R_t}(F^*) + \beta \cdot \sqrt{T}$$

where $F_t = \mathcal{A}(R_1, \dots, R_{t-1})$ and α, β are constants independent of T.

An online learning algorithm $\mathcal{A}$ selects the positions of the k facilities at each round $t \geq 1$ solely based on the positions of the clients in the previous rounds, $R_1, \dots, R_{t-1}$. If $\mathcal{A}$ is α-regret then Definition 3 implies that its time-averaged overall cost (connection + moving cost) is at most α times the time-averaged cost of the *optimal static solution!* [2] Furthermore, the dependency on $\sqrt{T}$ is known to be optimal [11] and β is typically only required to be polynomially bounded by the size of the input, as for $T \to \infty$ the corresponding term in the time-averaged cost vanishes.

As already mentioned, the seminal work of [11] implies the existence of an $(1+\epsilon)$-regret algorithm for Problem 1; however, this algorithm requires $\mathcal{O}(n^k)$ time and space complexity. Prior to this work, the only polynomial time online learning algorithm for Problem 1 was due to Fotakis et al. [30], for the special case of $\gamma = 0$. Specifically, in their work the authors design an online learning algorithm with the following guarantee:

Theorem (Fotakis et al. [30]). *There exists a randomized online learning algorithm for Problem 1 that runs in polynomial time (w.r.t. T, n and $\log D_G$) such that*

$$\mathbb{E}\left[\sum_{t=1}^{T} C_{R_t}(F_t)\right] \leq \mathcal{O}(k) \cdot \min_{|F^*|=k} \sum_{t=1}^{T} C_{R_t}(F^*) + \mathcal{O}(k \cdot n \cdot \sqrt{\log n} \cdot D_G) \cdot \sqrt{T}$$

Clearly, the algorithm of [30] has not been designed to account for charging the moving of facilities, as indicated by the absence of the moving cost in the above regret guarantee. The main contribution of this work is to obtain (for the first time) regret guarantees that also account for the moving cost.

Theorem 1. *There exists a randomized online learning algorithm for Problem 1 (Algorithm 2) that runs in polynomial time (w.r.t. T, n and $\log D_G$) and admits the following regret guarantee:*

$$\mathbb{E}\left[\sum_{t=1}^{T} C_{R_t}(F_t) + \gamma \cdot \sum_{t=2}^{T} M_G(F_{t-1}, F_t)\right] \leq \mathcal{O}(\log n) \cdot \min_{|F^*|=k} \sum_{t=1}^{T} C_{R_t}(F^*) + \beta \cdot \sqrt{T}$$

for $\beta = \mathcal{O}(k \cdot n^{3/2} \cdot D_G \cdot \max(\gamma, 1))$ and any $\gamma \geq 0$.

Remark 1. *We remark that while our additive regret β is larger than the corresponding term in [30] by a factor of $o(\sqrt{n})$, our results apply to any $\gamma \geq 0$ while the algorithm of [30] can generally suffer unbounded moving cost for $\gamma \to \infty$, as our experimental results verify.*

2.1 HSTs and Metric Embeddings

In this section we provide some preliminary introduction to Hierarchical Separation Trees (HSTs), as they consist a key technical tool towards proving Theorem 1. A *weighted tree* $\mathcal{T}(V, E, w)$ is a weighted graph with no cycles. Equivalently, for any pair of vertices $i, j \in V$ there exists a unique path that connects them. In Definition 4, we establish some basic notation for tree graphs.

Definition 4. *Fix any tree $\mathcal{T}(V, E, w)$. For every vertex $u \in V$, $\mathrm{cld}(u) \subseteq V$ denotes the set children vertices of u and $p(u)$ denotes its unique parent, i.e. $u \in \mathrm{cld}(p(u))$. The root $r \in V$ of $\mathcal{T}$ is the unique node with $p(r) = \varnothing$ and the set $L(\mathcal{T}) := \{u \in V : \mathrm{cld}(u) = \varnothing\}$ denotes the leaves of $\mathcal{T}$. We use $\mathrm{dpt}(u)$ to denote the depth of a vertex $u \in V$, i.e. the length of the (unique) path from the root r to u, and $h(\mathcal{T}) := \max_{u \in L(\mathcal{T})} \mathrm{dpt}(u)$ to denote the height of $\mathcal{T}$. We use $\mathrm{lev}(u) := h(\mathcal{T}) - \mathrm{dpt}(u)$ to denote the level of a vertex $u \in V$. Finally, $T(u) \subseteq V$ denotes the set of vertices on the sub-tree rooted at u, i.e. the set of vertices that are descendants of u.*

Next, we proceed to define a family of well-structured tree graphs that constitute one of the primary technical tools used in our analysis.

Definition 5. *A Hierarchical Separation Tree (HST) is a weighted tree $\mathcal{T}(V, E, w)$ such that (i) for any node u and any of its children $v \in cld(u)$, the edge $e = (u, v)$ admits weight $w_e = 2^{\mathrm{lev}(v)}$, and (ii) the tree is balanced, namely $lev(u) = 0$ for all leaves $u \in L(\mathcal{T})$.*

In their seminal works, [10] and later [24] showed that HSTs can approximately preserve the distances of any graph $G(V, E, w)$ within some logarithmic level of distortion.

[2] Specifically, the time-averaged overall cost of $\mathcal{A}$ approaches this upper bound with rate $\beta \cdot T^{-1/2}$.

Theorem 2. *For any graph $G(V,E,w)$ with $|V| = n$ and diameter D, there exists a polynomial-time randomized algorithm that given as input G produces an HST $\mathcal{T}$ with height $h(\mathcal{T}) \leq \lceil \log D \rceil$ s.t.*

1. *$L(\mathcal{T}) = V$, meaning that the leaves of $\mathcal{T}$ correspond to the vertices of G.*
2. *For any $u, v \in V$, $d_G(u,v) \leq d_{\mathcal{T}}(u,v)$ and $\mathbb{E}[d_{\mathcal{T}}(u,v)] \leq \mathcal{O}(\log n) \cdot d_G(u,v)$.*

Theorem 2 states that any weighted graph $G(V,E,w)$ can be embedded into an HST $\mathcal{T}$ with $\mathcal{O}(\log n)$-distortion. This means that the distance $d_G(u,v)$ between any pair of vertices $u, v \in V$ can be approximated by their respective distance $d_{\mathcal{T}}(u,v)$ in $\mathcal{T}$ within an (expected) factor of $\mathcal{O}(\log n)$.

Remark 2. *We note that traditionally HSTs are neither balanced nor are required to have weights that are specifically powers of 2. However, we can transform any general HST into our specific definition, and this has been accounted for in the statement of the above theorem. The details are deferred to Section B of the Appendix.*

3 Overview of our approach

In this section we present the key steps of our approach towards designing the $\mathcal{O}(\log n)$-regret online learning algorithm for Problem 1. Our approach can be summarized in the following three pillars:

1. In Section 3.1 we introduce a *fractional relaxation* of Problem 1 in the special case of HSTs (Problem 2). Problem 2 is an artificial problem at which the learner can place a *fractional amount of facility* to the leaves of an HST so as to fractionally serve the arrived clients. Since the *optimal static solution* of Problem 2 lower bounds the *optimal static solution* of Problem 1 in the special case of HSTs, the first step of our approach is to design an $\mathcal{O}(1)$-regret algorithm for Problem 2.
2. In Section 3.2 we present the formal guarantees of a novel randomized rounding scheme, called $\mathrm{Cut\&Round}$, that is client-oblivious and converts any *fractional solution* for Problem 2 into an actual placement of k facilities on the leaves of the HST with just an $\mathcal{O}(1)$-overhead in the connection and the moving cost.
3. In Section 3.3 we present how the *fractional algorithm* for Problem 2 together with the $\mathrm{Cut\&Round}$ rounding naturally lead to an $\mathcal{O}(1)$-regret online learning algorithm for Problem 1 in the special case of HSTs (Algorithm 1). Our main algorithm, presented in Algorithm 2, then consists of running Algorithm 1 into an $\mathcal{O}(\log n)$ HST embedding of input graph.

3.1 A Fractional Relaxation for HSTs

In this section we introduce a fractional relaxation for Problem 1, called *Fractional k-Clustering with Moving Costs on HSTs* (Problem 2). Fix any HST $\mathcal{T}(V,E,w)$ (in this section, V denotes the nodes of the HST). We begin by presenting a *fractional extension* of placing k facilities on the leaves of $\mathcal{T}$.

Definition 6. *The set of fractional facility placements $\mathcal{FP}(\mathcal{T})$ consists of all vectors $y \in \mathbb{R}^{|V|}$ such that*

1. $y_v \in [0,1]$ *for all leaves $v \in L(\mathcal{T})$.*
2. $y_v = \sum_{u \in \mathrm{cld}(v)} y_u$ *for all non-leaves $v \notin L(\mathcal{T})$.*
3. $\sum_{v \in L(\mathcal{T})} y_v = k$*, i.e. the total amount of facility on the leaves equals k.*

For a leaf vertex $v \in L(\mathcal{T})$, y_v simply denotes the fractional amount of facilities that are placed on it. For all non-leaf vertices $v \notin L(\mathcal{T})$, y_v denotes the total amount of facility placed in the leaves of the sub-tree $T(v)$. Thus, any integral vector $y \in \mathcal{FP}(\mathcal{T}) \cap \mathbb{N}$ corresponds to a placement of k facilities on the leaves of $\mathcal{T}$.

In Definitions 7 and 8 we extend the notion of connection and moving cost for fractional facility placements. In the special case of integral facility placements, Definitions 7 and 8 respectively collapse to Definitions 1 and 2 (a formal proof is given in Claims 1 and 2 of Section C of the Appendix).

Definition 7. *The fractional connection cost of a set of clients $R \subseteq L(\mathcal{T})$ under $y \in \mathcal{FP}(\mathcal{T})$ is defined as*

$$f_R(y) := \sum_{j \in R} \sum_{v \in P(j,r)} 2^{lev(v)+1} \cdot \max(0, 1 - y_v)$$

where $P(j,r)$ denotes the set of vertices in the (unique) path from the leaf $j \in L(\mathcal{T})$ to the root r.

Definition 8. *The fractional moving cost between any $y, y' \in \mathcal{FP}(\mathcal{T})$ is defined as*

$$||y - y'||_{\mathcal{T}} := \gamma \cdot \sum_{v \in V(\mathcal{T})} 2^{lev(v)} \cdot |y_v - y'_v|$$

We are now ready to present our fractional generalization of Problem 1 in the special case of HSTs.

Problem 2 (*Fractional k-Clustering with Moving Costs on HSTs*)**.** *Fix any HST $\mathcal{T}$. At each round $t = 1, \dots, T$:*

1. *The learner selects a vector $y^t \in \mathcal{FP}(\mathcal{T})$.*
2. *The adversary selects a set of clients $R_t \subseteq L(\mathcal{T})$.*
3. *The learner suffers cost $f_{R_t}(y^t) + ||y^t - y^{t-1}||_{\mathcal{T}}$.*

In Section 4, we develop and present an $\mathcal{O}(1)$-regret algorithm for Problem 2 (see Algorithm 3). To this end, we present its formal regret guarantee established in Theorem 3.

Theorem 3. *There exists a polynomial-time online learning algorithm for Problem 2 (Algorithm 3), such that for any sequence $R_1, \dots, R_T \subseteq L(\mathcal{T})$, its output $y^1, \dots, y^T$ satisfies*

$$\sum_{t=1}^{T} f_{R_t}(y^t) + \sum_{t=2}^{T} ||y^t - y^{t-1}||_{\mathcal{T}} \leq \frac{3}{2} \cdot \min_{y^* \in \mathcal{FP}(\mathcal{T})} \sum_{t=1}^{T} f_{R_t}(y^*) + \beta \cdot \sqrt{T}$$

for $\beta = \mathcal{O}\left(k \cdot |L(\mathcal{T})|^{3/2} \cdot D_{\mathcal{T}} \cdot \max(\gamma, 1)\right)$.

3.2 From Fractional to Integral Placements in HSTs

As already mentioned, the basic idea of our approach is to convert at each round $t \geq 1$ the *fractional placement* $y^t \in \mathcal{FP}(\mathcal{T})$ produced by Algorithm 3 into an integral facility placement $F_t \subseteq L(\mathcal{T})$ with $|F_t| = k$ on the leaves of the HST. In order to guarantee small regret, our rounding scheme should preserve both the connection and the moving cost of the fractional solution within constant factors for *any possible set of arriving clients*. In order to guarantee the latter, our rounding scheme $\mathrm{Cut\&Round}$ (Algorithm 4) uses shared randomness across different rounds. $\mathrm{Cut\&Round}$ is rather complicated and is presented in Section 5. To this end, we present its formal guarantee.

Theorem 4. *There exists a linear-time deterministic algorithm, called* $\mathrm{Cut\&Round}$ *(Algorithm 4), that takes as input an HST $\mathcal{T}$, a fractional facility placement $y \in \mathcal{FP}(\mathcal{T})$ and a vector $\alpha \in [0,1]^{|V|}$ and outputs a placement of k facilities $F \leftarrow \mathrm{Cut\&Round}(\mathcal{T}, y, \alpha)$ on the leaves of $\mathcal{T}$ ($F \subseteq L(\mathcal{T})$ and $|F| = k$) such that*

1. $\mathrm{E}_{\alpha \sim \mathrm{Unif}(0,1)}\left[C_R(F)\right] = f_R(y)$ *for all client requests $R \subseteq L(\mathcal{T})$.*
2. $\mathrm{E}_{\alpha \sim \mathrm{Unif}(0,1)}\left[\gamma \cdot M_{\mathcal{T}}(F, F')\right] \leq 4 \cdot ||y - y'||_{\mathcal{T}}$ *for all other fractional facility placements $y' \in \mathcal{FP}(\mathcal{T})$ and $F' \leftarrow \mathrm{Cut\&Round}(\mathcal{T}, y', \alpha)$.*

Item 1 of Theorem 4 establishes that although $\mathrm{Cut\&Round}$ is *oblivious* to the arrived set of clients $R_t \subseteq L(\mathcal{T})$, the expected connection cost of the output equals the *fractional connection cost* under $y^t \in \mathcal{FP}(\mathcal{T})$. Item 2 of Theorem 4 states that once the same random seed α is used into two consecutive time steps, then the expected moving cost between the facility-placements F_t and F_{t+1} is at most $\mathcal{O}(1)$-times the fractional moving cost between y^t and y^{t+1}. Both properties crucially rely on the structure of the HST and consist one of the main technical contributions of our work.

3.3 Overall Online Learning Algorithm

We are now ready to formally introduce our main algorithm (Algorithm 2) and prove Theorem 1. First, we combine the algorithms from Theorems 3 and 4 to design an $\mathcal{O}(1)$-regret algorithm for Problem 1 on HSTs (Algorithm 1). Up next we present how Algorithm 1 can be converted into an $\mathcal{O}(\log n)$-regret online learning algorithm for general graphs, using the metric embedding technique of Theorem 2, resulting to our final algorithm (Algorithm 2).

Algorithm 1 $\mathcal{O}(1)$-regret for HSTs.

```
1: Input: A sequence R_1, ..., R_T ⊆ L(𝒯).
2: The learner samples α_v ~ Unif(0,1) for all v ∈ V(𝒯).
3: for each round t = 1 to T do
4:     The learner places the k facilities to the leaves of the HST 𝒯 based on the output F_t := Cut&Round(𝒯, y^t, α).
5:     The learner learns R_t ⊆ L(𝒯).
6:     The learner updates y^{t+1} ∈ 𝓕𝓟(𝒯) by running Algorithm 3 for Problem 2 with input R_1, ..., R_t.
7: end for
```

Algorithm 2 $\mathcal{O}(\log n)$-regret for graphs.

```
1: Input: A sequence R_1, ..., R_T ⊆ L(𝒯).
2: The learner embeds G(V, E, w) into a (random) HST 𝒯 with L(𝒯) = V via the procedure of Theorem 2.
3: for each round t = 1 to T do
4:     The learner selects a facility-placement F_t ⊆ V.
5:     The learner learns R_t ⊆ V.
6:     The learner updates F_{t+1} by giving as input R_1, ..., R_t ⊆ L(𝒯) to Algorithm 1 for 𝒯.
7: end for
```

Theorem 5. *For any sequence of client requests $R_1, \ldots, R_T \subseteq L(\mathcal{T})$, the sequence of facility-placements $F_1, \ldots, F_T \subseteq L(\mathcal{T})$ produced by Algorithm 1 satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T} C_{R_t}(F_t) + \gamma \cdot \sum_{t=2}^{T} M_{\mathcal{T}}(F_t, F_{t-1})\right] \leq 6 \cdot \min_{|F^*|=k} \sum_{t=1}^{T} C_{R_t}(F^*) + \beta \cdot \sqrt{T}$$

for $\beta = \mathcal{O}\left(k \cdot |L(\mathcal{T})|^{3/2} \cdot D_{\mathcal{T}} \cdot \max(\gamma, 1)\right)$.

Theorem 5 establishes that Algorithm 1 achieves constant regret in the special case of HSTs and its proof easily follows by Theorems 3 and 4. Then, the proof of Theorem 1 easily follows by Theorem 2 and Theorem 5. All the proofs are deferred to Section C of the Appendix.

4 $\mathcal{O}(1)$-Regret for Fractional HST Clustering

In this section we present the $\mathcal{O}(1)$-regret algorithm for Problem 2, described in Algorithm 3, and exhibit the key ideas in establishing Theorem 3. Without loss of generality, we can assume that the facility-weight satisfies $\gamma \geq 1$[3].

Algorithm 3 is the well-known online learning algorithm *Follow the Regularized Leader* (FTRL) with a specific regularizer $R_{\mathcal{T}}(\cdot)$ presented in Definition 9. Our results crucially rely on the properties of this regularizer since it takes into account the HST structure and permits us to bound the fractional moving cost of FTRL.

Definition 9. *Given an HST $\mathcal{T}$, the dilated entropic regularizer $R_{\mathcal{T}}(y)$ over $y \in \mathcal{FP}(\mathcal{T})$ is defined as*

$$R_{\mathcal{T}}(y) := \sum_{v \neq r} 2^{\mathrm{lev}(v)} \cdot (y_v + \delta_v) \cdot \ln\left(\frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}\right)$$

where $\delta_v := (k/n) \cdot |L(\mathcal{T}) \cap T(v)|$ and $n := |L(\mathcal{T})|$.

Algorithm 3 selects at each step t the facility placement $y^t \in \mathcal{FP}(\mathcal{T})$ that minimizes a convex combination of the total fractional connection cost for the sub-sequence $R_1, \ldots, R_{t-1}$ and $R_{\mathcal{T}}(y)$. The regularization term ensures the stability of the output, which will result in a bounded fractional moving cost.

[3] If not, establishing our guarantees for $\gamma = 1$ will clearly upper bound the actual moving cost.

Algorithm 3 FTRL with dilated entropic regularization

1: **Input:** An adversarial sequence $R_1, \dots, R_T \subseteq L(\mathcal{T})$.
2: **for** $t = 1$ **to** T **do**
3: The learner selects $y^t \in \mathcal{FP}(\mathcal{T})$.
4: The learner suffers cost $f_{R_t}(y^t) + ||y^t - y^{t-1}||_{\mathcal{T}}$.
5: The learner updates $y^{t+1} \leftarrow \arg\min_{y \in \mathcal{FP}(\mathcal{T})} \left[\sum_{s=1}^{t} f_{R_s}(y) + (\gamma\sqrt{nT}) \cdot R_{\mathcal{T}}(y)\right]$.
6: **end for**

Analysis of Algorithm 3. Due to space limitations, all proofs are moved to Section D of the Appendix. The primary reason for the specific selection of the regularizer at Definition 9 is that $R_{\mathcal{T}}(\cdot)$ is strongly convex with respect to the norm $||\cdot||_{\mathcal{T}}$ of Definition 8, as established in Lemma 1 which is the main technical contribution of the section. We use $D = D_{\mathcal{T}}$ for the diameter of $\mathcal{T}$.

Lemma 1. *For any vectors $y, y' \in \mathcal{FP}(\mathcal{T})$,*

$$R_{\mathcal{T}}(y') \geq R_{\mathcal{T}}(y) + \langle \nabla R_{\mathcal{T}}(y), y' - y\rangle + \left(8kD\gamma^2\right)^{-1} \cdot ||y - y'||^2_{\mathcal{T}}$$

The strong convexity of $R_{\mathcal{T}}(y)$ with respect to $||\cdot||_{\mathcal{T}}$ is crucial since it permits us to bound the moving cost of Algorithm 3 by its fractional connection cost.

Lemma 2. *For any sequence $R_1, \dots, R_T \subseteq L(\mathcal{T})$, the output of Algorithm 3 satisfies*

$$\sum_{t=2}^{T} ||y^t - y^{t-1}||_{\mathcal{T}} \leq \frac{1}{2} \cdot \sum_{t=1}^{T} f_{R_t}(y^t) + \mathcal{O}\left(\gamma k D\right) \cdot \sqrt{T}$$

We remark that using another regularizer $R(\cdot)$ that is strongly convex with respect to another norm $||\cdot||$ would still imply Lemma 1 with respect to $||\cdot||$. The problem though is that the *fractional moving cost* $\sum_{t=1}^{T} ||y_t - y_{t-1}||$ can no longer be associated with the actual moving cost $\sum_{t=1}^{T} M_{\mathcal{T}}(F_t, F_{t-1})$. It is for this reason that using a regularizer that is strongly convex with respect to $||\cdot||_{\mathcal{T}}$ is crucial.

Next, by adapting the standard analysis of FTRL to our specific setting, we derive Lemma 3 establishing that Algorithm 3 admits bounded connection cost.

Lemma 3. *For any sequence $R_1, \dots, R_T \subseteq L(\mathcal{T})$, the output of Algorithm 3 satisfies*

$$\sum_{t=1}^{T} f_{R_t}(y^t) \leq \min_{y^* \in \mathcal{FP}} \sum_{t=1}^{T} f_{R_t}(y^*) + \mathcal{O}\left(kn^{3/2}D\gamma\right) \cdot \sqrt{T}$$

The proof of Theorem 3 directly follows by Lemma 2 and 3. We conclude the section by presenting how Step 5 of Algorithm 3 can be efficiently implemented, namely

$$\min_{y \in \mathcal{FP}(\mathcal{T})} \Phi_t(y) := \sum_{s=1}^{t} f_{R_s}(y) + (\gamma\sqrt{nT}) \cdot R_{\mathcal{T}}(y).$$

Since $\Phi_t(y)$ is strongly convex and the set $\mathcal{FP}(\mathcal{T})$ is a polytope, one could use standard optimization algorithms such as the *ellipsoid method* or *projected gradient descent* to approximately minimize $\Phi_t(y)$ given access to a *sub-gradient oracle for* $\Phi_t(\cdot)$. In Claim 11 of Section D of the Appendix, we establish that the sub-gradients of $\Phi(\cdot)$ can be computed in polynomial time and thus any of the previous methods can be used to approximately minimize $\Phi(\cdot)$. In Lemma 4 we establish the intuitive fact that approximately implementing Step 5 does not affect the guarantees of Theorem 3.

Lemma 4. *Let y^t be the minimizer of $\Phi_t(\cdot)$ in $\mathcal{FP}(T)$ and let $z^t \in \mathcal{FP}(\mathcal{T})$ be any point such that $\Phi_t(z^t) \leq \Phi_t(y^t) + \epsilon$ for some $\epsilon = \mathcal{O}(T^{-1/2})$. Then,*

$$f_{R_t}(z^t) + ||z^t - z^{t-1}||_{\mathcal{T}} \leq f_{R_t}(y^t) + ||y^t - y^{t-1}||_{\mathcal{T}} + \mathcal{O}\left(kn^{3/2}D\gamma\right) \cdot T^{-1/2}$$

Remark 3. *In our implementation of the algorithm, we approximately solve Step 5 of Algorithm 3 via Mirror Descent based on the Bregman divergence of $\mathcal{R}_{\mathcal{T}}(\cdot)$. This admits the same convergence rates as projected gradient descent but the projection step can be computed in linear time with respect to the size of the HST $\mathcal{T}$. We present the details of our implementation in Section C of the Appendix.*

5 The Cut&Round Rounding

In this section we present our novel rounding scheme (Algorithm Cut&Round) as well as the main steps that are required in order to establish Theorem 4. To ease notation, for any real number $x \geq 0$ we denote its decimal part as $\delta(x) = x - \lfloor x \rfloor$.

Algorithm 4 Cut&Round.

1: **Input:** An HST $\mathcal{T}$, a fractional placement $y \in \mathcal{FP}(\mathcal{T})$ and thresholds $\alpha_v \in [0,1]$ for all $v \in V(\mathcal{T})$.
2: $Y_r \leftarrow k$
3: **for** levels $\ell = h(\mathcal{T})$ **to** 1 **do**
4:   **for** all nodes v with $lev(v) = \ell$ **do**
5:     $Y_{rem} \leftarrow Y_v$
6:     $y_{rem} \leftarrow y_v$
7:     **for** all children $u \in \mathrm{cld}(v)$ **do**
8:       $Y_u \leftarrow \mathrm{Alloc}(y_u, Y_{rem}, y_{rem}, \alpha_u)$
9:       $Y_{rem} \leftarrow Y_{rem} - Y_u$
10:       $y_{rem} \leftarrow y_{rem} - y_u$
11:     **end for**
12:   **end for**
13: **end for**
14: **return** $F := \{u \in L(\mathcal{T}) : Y_u = 1\}$.

Algorithm 5 Alloc.

Input: Numbers $y_u, y_{rem} \geq 0, Y_{rem} \in \mathbb{N}$ and $\alpha_u \in [0,1]$.
if $Y_{rem} == \lfloor y_{rem} \rfloor$ **then**
  if $\delta(y_u) < \delta(y_{rem})$ **then**
    $Y_u \leftarrow \lfloor y_u \rfloor$
  else
    $Y_u \leftarrow \lfloor y_u \rfloor + \mathbb{1}\left[a_u \leq \frac{\delta(y_u) - \delta(y_{rem})}{1 - \delta(y_{rem})}\right]$
  end if
else
  if $\delta(y_u) < \delta(y_{rem})$ **then**
    $Y_u \leftarrow \lfloor y_u \rfloor + \mathbb{1}\left[a_u \leq \frac{\delta(y_u)}{\delta(y_{rem})}\right]$
  else
    $Y_u \leftarrow \lfloor y_u \rfloor + 1$
  end if
end if
Return Y_u.

On principle, Cut&Round (Algorithm 4) assigns to each vertex v an integer number of facilities Y_v to be placed at the leaves of its sub-tree. Notice that due to sub-routine Alloc (Algorithm 5), Y_v either equals $\lfloor y_v \rfloor$ or $\lfloor y_v \rfloor + 1$. Cut&Round initially assigns k facilities to the set of leaves that descend from the root r, which is precisely $L(\mathcal{T})$. Then, it moves in decreasing level order to decide Y_v for each node v. Once Y_v is determined (Step 5), the Y_v facilities are allocated to the sub-trees of its children $u \in \mathrm{cld}(v)$ (Steps 7-10) via sub-routine Alloc using the thresholds α_u, in a manner that guarantees that $Y_v = \sum_{u \in \mathrm{cld}(v)} Y_u$ (see Section E.1 of the Appendix). This implies the feasibility of Cut&Round, as exactly k facilities are placed in the leaves of $\mathcal{T}$ at the end of the process.

Assuming that the set of thresholds α_v is randomly drawn from the uniform distribution in $[0,1]$, sub-routine Alloc (Algorithm 5) guarantees that Y_v either equals $\lfloor y_v \rfloor$ or $\lfloor y_v \rfloor + 1$ while $\mathbb{E}_\alpha[Y_v] = y_v$. This is formally captured in Lemma 5 and is crucial in the proof of Theorem 4.

Lemma 5. *Consider Algorithm 4 given as input a vector $y \in \mathcal{FP}(\mathcal{T})$ and random thresholds $\alpha_v \sim \mathrm{Unif}(0,1)$. Then,*

$$Y_v = \begin{cases} \lfloor y_v \rfloor & \text{with probability } 1 - \delta(y_v) \\ \lfloor y_v \rfloor + 1 & \text{with probability } \delta(y_v) \end{cases}$$

By coupling Lemma 5 with the HST structure we are able to establish Theorem 4. The proof is technically involved and thus deferred to Section E of the Appendix.

6 Conclusion

In this work, we designed the first polynomial-time online learning algorithm for *Online k-Clustering with Moving Costs* that achieves $\mathcal{O}(\log n)$-regret with respect to the cost of the optimal *static* facility placement, extending the results of Fotakis et al. [30] for the special case of $\gamma = 0$. A interesting future direction is to investigate whether a polynomial-time online learning algorithm with $\mathcal{O}(1)$-regret for the problem is theoretically possible or not.

Limitations: Our current optimality guarantees are with respect to the optimal *static* facility placement. Going beyond the notion of regret, an intriguing future direction is establishing guarantees with respect to the *optimal dynamic facility-placement* that moves facilities from round to round by suffering the corresponding moving cost.

References

[1] Alekh Agarwal, Daniel J. Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, 2014.

[2] Nir Ailon. Improved bounds for online learning over the permutahedron and other ranking polytopes. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, AISTATS 2014, 2014.

[3] Soroush Alamdari and David B. Shmoys. A bicriteria approximation algorithm for the k-center and k-median problems. In *Workshop on Approximation and Online Algorithms*, 2017.

[4] Hyung-Chan An, Ashkan Norouzi-Fard, and Ola Svensson. Dynamic facility location via exponential clocks. *ACM Trans. Algorithms*, 13(2):21:1–21:20, 2017.

[5] Tarique Anwar, Surya Nepal, Cecile Paris, Jian Yang, Jia Wu, and Quan Z. Sheng. Tracking the evolution of clusters in social media streams. *IEEE Transactions on Big Data*, pages 1–15, 2022.

[6] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, page 21–29. Association for Computing Machinery, 2001.

[7] Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *J. Comput. Syst. Sci.*, 2008.

[8] Maria-Florina Balcan and Avrim Blum. Approximation algorithms and online mechanisms for item pricing. In *ACM Conference on Electronic Commerce*, 2006.

[9] Nikhil Bansal, Niv Buchbinder, Aleksander Madry, and Joseph Naor. A polylogarithmic-competitive algorithm for the k-server problem. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 267–276. IEEE Computer Society, 2011.

[10] Yair Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. *Proceedings of 37th Conference on Foundations of Computer Science*, pages 184–193, 1996.

[11] Avrim Blum and Carl Burch. On-line learning and the metrical task system problem. *Mach. Learn.*, 39(1):35–58, 2000.

[12] Sébastien Bubeck, Michael B. Cohen, Yin Tat Lee, James R. Lee, and Aleksander Madry. k-server via multiscale entropic regularization. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 3–16. ACM, 2018.

[13] Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99. IEEE Computer Society, 1999.

[14] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem (extended abstract). In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, STOC '99, page 1–10. Association for Computing Machinery, 1999.

[15] Moses Charikar and Shi Li. A dependent lp-rounding approach for the k-median problem. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012*, volume 7391 of *Lecture Notes in Computer Science*, pages 194–205. Springer, 2012.

[16] Vincent Cohen-Addad, Benjamin Guedj, Varun Kanade, and Guy Rom. Online k-means clustering. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 1126–1134. PMLR, 2021.

[17] Aaron Cote, Adam Meyerson, and Laura J. Poplawski. Randomized k-server on hierarchical binary trees. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 227–234. ACM, 2008.

[18] Bart de Keijzer and Dominik Wojtczak. Facility reallocation on the line. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 188–194, 2018.

[19] Sina Dehghani, Soheil Ehsani, MohammadTaghi Hajiaghayi, Vahid Liaghat, and Saeed Seddighin. Stochastic k-server: How should uber work? In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 126:1–126:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[20] Sina Dehghani, MohammadTaghi Hajiaghayi, Hamid Mahini, and Saeed Seddighin. Price of competition and dueling games. *arXiv preprint arXiv:1605.04004*, 2016.

[21] Miroslav Dudík, Nika Haghtalab, Haipeng Luo, Robert E. Schapire, Vasilis Syrgkanis, and Jennifer Wortman Vaughan. Oracle-efficient online learning and auction design. In *58th IEEE Annual Symposium on Foundations of Computer Science*, FOCS 2017, 2017.

[22] Miroslav Dudík, Daniel J. Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI 2011, 2011.

[23] David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility location in evolving metrics. In *Automata, Languages, and Programming - 41st International ColloquiumICALP 2014*, volume 8573 of *Lecture Notes in Computer Science*, pages 459–470. Springer, 2014.

[24] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 448–455, New York, NY, USA, 2003. Association for Computing Machinery.

[25] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004.

[26] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Optimistic regret minimization for extensive-form games via dilated distance-generating functions. In *Neural Information Processing Systems*, 2019.

[27] Hendrik Fichtenberger, Silvio Lattanzi, Ashkan Norouzi-Fard, and Ola Svensson. Consistent k-clustering for general metrics. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2660–2678. SIAM, 2021.

[28] Dimitris Fotakis, Loukas Kavouras, Panagiotis Kostopanagiotis, Philip Lazos, Stratis Skoulakis, and Nikos Zarifis. Reallocating multiple facilities on the line. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, pages 273–279, 2019.

[29] Dimitris Fotakis, Thanasis Lianeas, Georgios Piliouras, and Stratis Skoulakis. Efficient online learning of optimal rankings: Dimensionality reduction via gradient descent. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020.

[30] Dimitris Fotakis, Georgios Piliouras, and Stratis Skoulakis. Efficient online learning for dynamic k-clustering. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3396–3406. PMLR, 18–24 Jul 2021.

[31] Takahiro Fujita, Kohei Hatano, and Eiji Takimoto. Combinatorial online prediction via metarounding. In *24th International Conference on Algorithmic Learning Theory*, ALT 2013, 2013.

[32] Dan Garber. Efficient online linear optimization with approximation algorithms. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS 2017, 2017.

[33] Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. Consistent k-median: Simpler, better and robust. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1135–1143. PMLR, 13–15 Apr 2021.

[34] Elad Hazan. Introduction to online convex optimization. *CoRR*, abs/1909.05207, 2019.

[35] Elad Hazan, Wei Hu, Yuanzhi Li, and Zhiyuan Li. Online improper learning with an approximation oracle. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[36] Elad Hazan and Satyen Kale. Online submodular minimization. *J. Mach. Learn. Res.*, 2012.

[37] Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. In *25th Annual Conference on Learning Theory*, COLT 2012, 2012.

[38] Elad Hazan and Tomer Koren. The computational power of optimization in online learning. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, STOC 2016, 2016.

[39] David P. Helmbold, Robert E. Schapire, and M. Long. Predicting nearly as well as the best pruning of a decision tree. In *Machine Learning*, 1997.

[40] David P. Helmbold and Manfred K. Warmuth. Learning permutations with exponential weights. In *Proceedings of the 20th Annual Conference on Learning Theory*, COLT 2007, 2007.

[41] Nicole Immorlica, Adam Tauman Kalai, Brendan Lucier, Ankur Moitra, Andrew Postlewaite, and Moshe Tennenholtz. Dueling algorithms. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 215–224, New York, NY, USA, 2011. Association for Computing Machinery.

[42] Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 731–740. ACM, 2002.

[43] Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.

[44] Stefanie Jegelka and Jeff A. Bilmes. Online submodular minimization for combinatorial structures. In *Proceedings of the 28th International Conference on Machine Learning*, ICML 2011, 2011.

[45] Sham Kakade, Adam Tauman Kalai, and Katrina Ligett. Playing games with approximation algorithms. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, STOC 2007, 2007.

[46] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. In *J. Comput. Syst. Sci*. Springer, 2003.

[47] Wouter M. Koolen, Manfred K. Warmuth, and Jyrki Kivinen. Hedging structured concepts. In *the 23rd Conference on Learning Theory, COLT 2010*, 2010.

[48] Elias Koutsoupias. The k-server problem. *Comput. Sci. Rev.*, 3(2):105–118, 2009.

[49] Elias Koutsoupias and Christos H. Papadimitriou. On the k-server conjecture. *J. ACM*, 42(5):971–983, 1995.

[50] Amit Kumar. Constant factor approximation algorithm for the knapsack median problem. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, page 824–832. Society for Industrial and Applied Mathematics, 2012.

[51] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM*, 57(2), 2010.

[52] Silvio Lattanzi and Sergei Vassilvitskii. Consistent k-clustering. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1975–1984. PMLR, 2017.

[53] Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. *SIAM J. Comput.*, 45(2):530–547, 2016.

[54] Jyh-Han Lin and Jeffrey Scott Vitter. Approximation algorithms for geometric median problems. *Information Processing Letters*, 44(5):245 – 249, 1992.

[55] M.E.J. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[56] Romualdo Pastor-Satorras and Alessandro Vespignani. Epidemic Spreading in Scale-Free Networks. *Physical Review Letters*, 86(14):3200–3203, 2001.

[57] Holakou Rahmanian and Manfred K. Warmuth. Online dynamic programming. In *NIPS*, 2017.

[58] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, and Philippe Vanhems. High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6(8), 2011.

[59] Matthew J. Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *22nd Annual Conference on Neural Information Processing Systems*, NIPS 2008, 2008.

[60] Daiki Suehiro, Kohei Hatano, Shuji Kijima, Eiji Takimoto, and Kiyohito Nagano. Online prediction under submodular constraints. In *Algorithmic Learning Theory*, ALT 2012, 2012.

[61] Eiji Takimoto and Manfred K. Warmuth. Predicting nearly as well as the best pruning of a planar decision graph. In *Theoretical Computer Science*, 2000.

[62] Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. *J. Mach. Learn. Res.*, 2003.

[63] Chayant Tantipathananandh, Tanya Berger-Wolf, and David Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 717–726. Association for Computing Machinery, 2007.

[64] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, USA, 1st edition, 2011.

[65] Shota Yasutake, Kohei Hatano, Shuji Kijima, Eiji Takimoto, and Masayuki Takeda. Online linear optimization over permutations. In *Proceedings of the 22nd International Conference on Algorithms and Computation*, ISAAC 2011, 2011.

[66] Neal E. Young. K-medians, facility location, and the chernoff-wald bound. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, page 86–95. Society for Industrial and Applied Mathematics, 2000.

Appendix

A Further Related Work

In this chapter of the appendix, we continue our discussion on the literature that relates to this work.

Efficient Combinatorial Online Learning. There exists a long line of research studying efficient online learning algorithms in various combinatorial domains (e.g., selection of paths, permutations, binary search trees etc.) [39, 61, 62, 40, 7, 59, 44, 22, 41, 65, 37, 36, 1, 2, 20, 29]. Another related line of work studies *black-box reductions* converting any α-approximation (offline) algorithm to an $\mathcal{O}(\alpha)$-regret online learning algorithm for a specific class of combinatorial optimization problems called *linear optimization problems* [46, 8, 45, 47, 60, 31, 38, 57, 21, 32, 35]. We remark that a key difference of our setting with the aforementioned works is that in the latter case the learner is not penalized for switching actions from round to round with an additional moving/switching cost. In the context of Problem 1 this means that $\gamma = 0$ which is exactly the setting considered by [30]. As a result, apart from the fact that $k\text{-median}$ does not belong in the class of *linear optimization problems*, the aforementioned *black-box reductions* do not apply to Problem 1 since they do not account for the moving cost.

The k-server Problem. Our work also relates with the rich line of literature on the $k\text{-server}$ problem [49, 17, 48, 9, 19, 12]. In this setting there exists only 1 client at each round, while 1-lookahead is assumed, i.e. the request R_t is revealed prior to the action of the algorithm at step t. Moreover in $k\text{-server}$ a facility must be placed in the exact position of the request, leading to a simpler combinatorial structure with respect to Problem 1[4]. However, in the $k\text{-server}$ problem, instead of using the benchmark of *regret*, the more challenging metric of *competitive ratio* that measures the sub-optimality with respect to the *optimal dynamic solution* is used. Mostly related to ours is the work of [9] providing the first $\text{poly}(\log n)$-competitive algorithm for $k\text{-server}$ by reducing the problem to the special case of HSTs. [9] first design a $\text{poly}(\log n)$-competitive algorithm for a fractional version of $k\text{-server}$ at which facilities can be fractionally placed into the vertices of the HST. They then use a randomized rounding scheme to convert the fractional solution into an integral one. The basic difference of the randomized rounding scheme of [9] with the one that we introduce in this work (Algorithm $\text{Cut\&Round}$) is that the first provides guarantees only for the moving cost of the facilities while $\text{Cut\&Round}$ provides guarantees both for the moving cost of the facilities as well as the connection cost of the clients.

Consistent k-Clustering. Another setting of clustering in the presence of unknown clients is that of *Consistent k-Clustering* [52, 33, 27]. In this setting, given an *unknown stream of clients*, a set of k facilities has to be maintained over time so that at any round t, the selected facilities form an approximately optimal solution of the sub-instance consisting of clients appeared in the time interval $\{1, t\}$. A basic difference of Consistent k-Clustering with Problem 1 is that in the first case the moving cost is not penalized as long as the number of swaps does not exceed a certain threshold ($\mathcal{O}(k)$).

[4]Given offline access to the sequence of requests, the optimal solution for the $k\text{-server}$ can be computed in polynomial-time while the optimal static solution of Problem 1 cannot be approximated in polynomial-time with ratio less than $(1 + 2/e)$ even under *a-priori* knowledge of the request sequence (inapproximability of $k\text{-median}$).

B Proof of Theorem 2

In this chapter of the appendix we briefly discuss the details behind Theorem 2 and show how the results of [10] and [24] hold even for the specific definition of HSTs we have considered in Definition 5.

Traditionally, HSTs are not required to be balanced nor are required to have weights that are specifically powers of 2. In fact, the seminal work of [10], later improved by [24], states that there exists a randomized procedure such that for every weighted graph $G(V, E, w)$, it constructs (in polynomial-time) a tree $\mathcal{T}$ such that:

1. There exists a perfect matching $\sigma : V \mapsto L(\mathcal{T})$ that maps the vertices of G to the leaves of $\mathcal{T}$.
2. For any vertices $i, j \in V$, their corresponding distance on $\mathcal{T}$ can only increase, i.e. $d_G(i,j) \leq d_{\mathcal{T}}(\sigma(i), \sigma(j))$.
3. On expectation, distances between vertices are distorted only by a logarithmic factor, i.e. $\mathbb{E}\left[d_{\mathcal{T}}(\sigma(i), \sigma(j))\right] \leq \mathcal{O}(\log|V|) \cdot d_G(i,j)$
4. The weight of any edge $e = (v, u)$ between a vertex $v \in V(\mathcal{T})$ and its parent vertex u is precisely $diam(G) \cdot 2^{-dpt(v)}$.
5. The height of $\mathcal{T}$ satisfies $h(\mathcal{T}) \leq \lceil \log\left(diam(G)\right) \rceil$.

The purpose of this section is to argue that one can easily transform such a tree $\mathcal{T}$ to match our notion of HSTs (Definition 5), while maintaining the same guarantees for the distortion of the distances. Recall that we have already assumed that the minimum edge weight of G is 1, i.e. $\min_{e \in E} w_e = 1$. Furthermore, we can also assume without loss of generality that the diameter of G is a power of 2; if not, simple scaling arguments suffice to transform G into such a graph by only distorting distances by a constant factor. Thus, we assume that $diam(G) = 2^d$ for some $d \geq 0$.

We start from the tree $\mathcal{T}$ that the algorithm of [24] generates. Recall that by definition, the weight of an edge $e = (i, j)$ between some vertex i and its parent node j is $2^{d - dpt(i)}$. In order to balance the tree, we take each leaf vertex $u \in L(\mathcal{T})$ at depth $dpt(u)$ and extend it downwards by adding new vertices until it reaches a new depth $dpt'(u) = d$. For every new edge that we add during this process, we maintain that the weight of the edge $e = (i, j)$ from i to its parent j is $diam(G) \cdot 2^{-dpt(i)}$.

Let $\mathcal{T}'$ be used to denote our modified tree. Clearly, the above construction guarantees $h(\mathcal{T}') = d$. Since by definition $h(\mathcal{T}) \leq \lceil \log\left(diam(G)\right) \rceil = d$, we know that all leaves initially lied at depth at most d, and thus by the end of the above process all leaves will lie at the same level of the tree and have depth d. Thus, we have indeed constructed a balanced tree. Furthermore, since by definition $dpt(v) = h(\mathcal{T}) - lev(v)$, we get that the weight of the edge $e = (i, j)$ from i to its parent j is $w_e = diam(G) \cdot 2^{lev(i) - d} = 2^{lev(i)}$. So, the constructed tree indeed satisfies all the requirements of Definition 5 and is a valid HST (according to our definition).

We will now argue that $\mathcal{T}'$ also satisfies all items of Theorem 2. Fist of all, the height of our new tree is precisely d, and thus it is true that $h(\mathcal{T}') \leq \lceil \log\left(diam(G)\right) \rceil$. Furthermore, since we only added edges to the initial tree $\mathcal{T}$, the distance between any two leaves can only increase. Thus, we get that for any vertices $i, j \in V$ it holds

$$d_G(i,j) \leq d_{\mathcal{T}}(i,j) \leq d_{\mathcal{T}'}(i,j)$$

Finally, it remains to upper bound the expected distortion on $\mathcal{T}'$. Recall that by construction of [24], we know that

$$\mathbb{E}\left[d_{\mathcal{T}}(\sigma(i), \sigma(j))\right] \leq \mathcal{O}(\log|V|) \cdot d_G(i,j)$$

Since edge lengths decrease by a factor of 2 every time we move down the tree, we know that the total length of the path we added in order to move leaf i from depth $dpt(i)$ to depth d is precisely $1 + 2 + \dots 2^{dpt(i)-1} \leq 2^{dpt(i)}$. This implies that any distance on $\mathcal{T}'$ can be at most twice the corresponding distance on $\mathcal{T}$, i.e.

$$d_{\mathcal{T}'}(\sigma(i), \sigma(j)) \leq 2 \cdot d_{\mathcal{T}}(\sigma(i), \sigma(j))$$

which completes the proof.

C Proofs of Section 3

In this chapter of the appendix we present all the omitted proofs from Section 3 concerning the basic algorithmic primitives we use in order to establish our main result in Theorem 1.

Roadmap. In section C.1 we establish the connection between Problems 1 and 2 and show that our notion of fractional connection and moving cost collapses with our initial definitions in the case of integral facility placements. Then, in section C.2 we present the proof of Theorem 5 and in section C.3 we present the proof of Theorem 1.

C.1 Establishing the relation between Problems 1 and 2

Fix any HST $\mathcal{T}$ and let $\mathcal{FP}(\mathcal{T})$ be the corresponding set of fractional facility placements. In this section, we will establish that in the case of integral facility placements $y \in \mathcal{FP}(\mathcal{T}) \cap \mathbb{N}$, the notions of fractional connection cost and fractional moving cost (formally stated in Definitions 7 and 8) collapse to the notions of actual connection and moving costs (formally stated in Definitions 1 and 2) respectively.

Let $y \in \mathcal{FP}(\mathcal{T}) \cap \mathbb{N}$ be an integral facility placement. Then, by definition, for each leaf $v \in L(\mathcal{T})$ we have $y_v \in \{0,1\}$ facilities that are placed on it, and the total amount of placed facilities is k, i.e. $\sum_{v \in L(\mathcal{T})} y_v = k$. Thus, we can associate with any integral facility placement y a corresponding set

$$F(y) = \{v \in L(\mathcal{T}) : y_v = 1\}$$

such that $|F(y)| = k$, meaning that $F(y)$ is a valid facility placement of the leaves of the $\mathcal{T}$.

In Claim 1 we will establish that for any set of clients, the connection cost under $F(y)$ is equal to the fractional connection cost under y. Then, in Claim 2 we will establish that the fractional moving cost between y and y' gives us precisely the moving cost between facility placements $F(y)$ and $F(y')$ on $\mathcal{T}$.

Claim 1. *For any integral facility placement $y \in \mathcal{FP}(\mathcal{T}) \cap \mathbb{N}$ and any set of clients $R \subseteq L(\mathcal{T})$, it holds that*

$$f_R(y) = C_R(F(y))$$

Proof. Fix any $y \in \mathcal{FP}(\mathcal{T}) \cap \mathbb{N}$ and any $R \subseteq L(\mathcal{T})$. By definition of the connection cost (Definition 1), we have

$$C_R(F) = \sum_{j \in R} \min_{i \in F(y)} d_{\mathcal{T}}(i,j)$$

Let's fix a particular client that lies on some leaf $j \in L(\mathcal{T})$ of $\mathcal{T}$. Let $i^* = \arg\min_{i \in F(y)} d_{\mathcal{T}}(i,j)$ be the leaf closest to j that $F(y)$ places a facility into. Since $\mathcal{T}$ is an HST and distances increase by a factor of 2 as we move up the tree, it is not hard to see that i^* is the leaf in $F(y)$ whose *lowest common ancestor* (lca) with j has the smallest level. Let $l^* = lca(j, i^*)$. Equivalently, l^* is the minimum-level vertex in $P(j,r)$ such that $y_{l^*} \geq 1$. Since $\mathcal{T}$ is balanced, we have that the connection cost of client j under $F(y)$ is precisely

$$C_{\{j\}}(F(y)) = 2 \cdot d_{\mathcal{T}}(j, l^*) = 2 \cdot \sum_{l=0}^{\mathrm{lev}(l^*)-1} 2^l$$

and since by integrality we have that $y_v = 0$ for any $v \in P(j, l^*) \setminus \{l^*\}$ and $y_v \geq 1$ for all $v \in P(l^*, r)$, we have

$$C_{\{j\}}(F) = 2 \cdot \sum_{v \in P(j,r)} 2^{\mathrm{lev}(v)} \cdot \max(0, 1 - y_v)$$

Summing over all clients $j \in R$ we get

$$C_R(F(y)) = \sum_{j \in R} \sum_{v \in P(j,r)} 2^{\mathrm{lev}(v)+1} \cdot \max(0, 1 - y_v) = f_R(y)$$

which concludes the proof. □

Claim 2. *For any integral facility placements $y, y' \in \mathcal{FP}(\mathcal{T}) \cap \mathbb{N}$, it holds that*

$$||y - y'||_{\mathcal{T}} = \gamma \cdot M_{\mathcal{T}}(F(y), F(y'))$$

Proof. Fix any two integral facility placements $y, y' \in \mathcal{FP}(\mathcal{T}) \cap \mathbb{N}$. By definition of the moving cost (Definition 2), we have that

$$M_{\mathcal{T}}(F(y), F(y')) = \min_{\sigma \in \Sigma} \sum_{i \in F(y)} d_{\mathcal{T}}(i, \sigma(i))$$

where Σ is the set of all possible matchings from the facilities in $F(y)$ to the facilities in $F(y')$.

In general graphs, the minimum transportation cost can have a very complicated structure and typically requires solving a minimum transportation problem in order to compute it. However, in the special case of HSTs, we are actually able to obtain a very simple expression for this quantity.

Recall that in an HST $\mathcal{T}$, edge weights increase by a factor of 2 every time we move up a level on the tree. Thus, it is always in out interest to move facilities between leaves whose lowest common ancestor is as low as possible. In other words, the matching σ that minimizes the transportation cost from $F(y)$ to $F(y')$ can be obtained by selecting an arbitrary leaf in $F(y)$, matching it to the leaf in $F(y')$ with which it shares the *lowest* lowest common ancestor and then repeating the process for the rest of the leaves.

Now fix any vertex $v \in V(\mathcal{T})$. Recall that y_v is equal to the number of facilities in $F(y)$ that are placed in the descendant leaves of v (respectively for y'_v). Thus, if we apply the above (optimal) transportation plan, the number of facilities that will end up traversing the edge from v to its parent vertex is going to be precisely $|y_v - y'_v|$. Since the weight of this edge is by definition $2^{\text{lev}(v)}$, we get that

$$M_{\mathcal{T}}(F(y), F(y')) = \sum_{v \in V(\mathcal{T})} 2^{\text{lev}(v)} \cdot |y_v - y'_v|$$

and since

$$||y - y'||_{\mathcal{T}} = \gamma \cdot \sum_{v \in V(\mathcal{T})} 2^{\text{lev}(v)} \cdot |y_v - y'_v|$$

we have proven the claim.

$\square$

C.2 Proof of Theorem 5

We will now formally present the proof of Theorem 5, bounding the expected total cost of Algorithm 1. Fix any sequence of clients $R_1, \dots, R_T$. Since the random seed α is selected uniformly at random (Step 3 of Algorithm 1), by Item 1 of Theorem 4 we get that

$$\mathbb{E}\left[C_{R_t}(F_t)\right] = f_{R_t}(y^t)$$

Moreover since the same random seed α is used at all rounds $t \geq 1$, Item 2 of Theorem 4 implies that

$$\gamma \cdot \mathbb{E}\left[M_{\mathcal{T}}(F_{t+1}, F_t)\right] \leq 4 \cdot ||y^{t+1} - y^t||_{\mathcal{T}}$$

Thus,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} C_{R_t}(F_t) + \gamma \cdot \sum_{t=2}^{T} M_{\mathcal{T}}(F_t, F_{t-1})\right] &\leq 4 \cdot \left(\sum_{t=1}^{T} f_{R_t}(y^t) + \sum_{t=2}^{T} ||y^t - y^{t-1}||_{\mathcal{T}}\right) \\
&\leq 6 \cdot \min_{y^* \in \mathcal{FP}} \sum_{t=1}^{T} f_{R_t}(y^*) + \beta \cdot \sqrt{T}
\end{aligned}$$

where the last inequality follows by Theorem 3 for $\beta = \mathcal{O}\left(k \cdot |L(\mathcal{T})|^{3/2} \cdot D_{\mathcal{T}} \cdot \max(\gamma, 1)\right)$. The proof is concluded by the fact that

$$\min_{y^* \in \mathcal{FP}} \sum_{t=1}^{T} f_{R_t}(y^*) \leq \min_{|F^*|=k} \sum_{t=1}^{T} C_{R_t}(F^*)$$

which is established in Claim 1 of Appendix C.1, stating that for any placement of k-facilities $F \subseteq L(\mathcal{T})$ there exists a corresponding $y \in \mathcal{FP}(\mathcal{T})$ whose fractional connection cost is equal to F's under any client request.

C.3 Proof of Theorem 1

We will now formally present the proof of Theorem 1, bounding the regret of Algorithm 2.

Let $\mathcal{T}$ be the HST that we randomly embed our graph $G(V, E, w)$ into. Since $V = L(\mathcal{T})$, we slightly abuse notation and use u to refer both to some vertex of G and to the corresponding leaf of $\mathcal{T}$. From Theorem 5, we know that the output of Algorithm 1 satisfies

$$\mathbb{E}\left[\sum_{t=1}^{T} C_{R_t}^{\mathcal{T}}(F_t) + \gamma \cdot \sum_{t=2}^{T} M_{\mathcal{T}}(F_t, F_{t-1})\right] \leq 6 \cdot \min_{|F^*|=k} \sum_{t=1}^{T} C_{R_t}^{\mathcal{T}}(F^*) + \mathcal{O}\left(k \cdot |L(\mathcal{T})|^{3/2} \cdot D_{\mathcal{T}} \cdot \max(1, \gamma)\right) \cdot \sqrt{T}$$

where we use $\mathcal{T}$ in the connection and moving cost to indicate that all distances are measured on the HST. Here, the expectation is taken over the random choices of Algorithm 1.

Next, notice that both the connection cost and the moving cost are defined as sum of distances. Thus, the results of Theorem 2 about the distance distortion from G to $\mathcal{T}$ clearly apply for these quantities as well, namely

$$C_{R_t}^{G}(F_t) \leq C_{R_t}^{\mathcal{T}}(F_t) \text{ and } \mathbb{E}\left[C_{R_t}^{\mathcal{T}}(F_t)\right] \leq \mathcal{O}(\log|V|) \cdot C_{R_t}^{G}(F_t)$$

and

$$M_G(F_t, F_{t-1}) \leq M_{\mathcal{T}}(F_t, F_{t-1}) \text{ and } \mathbb{E}\left[M_{\mathcal{T}}(F_t, F_{t-1})\right] \leq \mathcal{O}(\log|V|) \cdot M_G(F_t, F_{t-1})$$

Thus, taking an expectation over the randomness of $\mathcal{T}$, we finally get that

$$\mathbb{E}\left[\sum_{t=1}^{T} C_{R_t}^{G}(F_t) + \gamma \cdot \sum_{t=2}^{T} M_G(F_t, F_{t-1})\right] \leq \mathcal{O}(\log|V|) \cdot \min_{|F^*|=k} \sum_{t=1}^{T} C_{R_t}^{G}(F^*) + \mathcal{O}\left(k \cdot |L(\mathcal{T})|^{3/2} \cdot D_{\mathcal{T}} \cdot \max(1, \gamma)\right) \cdot \sqrt{T}$$

Let $n = |V|$ and $D = diam(G)$. From the above, we get that Algorithm 2 is indeed α-regret for $\alpha = \mathcal{O}(\log n)$. Furthermore, we have that $|L(\mathcal{T})| = |V| = n$, and $D_{\mathcal{T}} = 2 \cdot (2^{h(\mathcal{T})} - 1) \leq 4D$ since $h(\mathcal{T}) \leq \lceil \log D \rceil$. Thus, setting $\beta = \mathcal{O}(k \cdot n^{3/2} \cdot D \cdot \max(1, \gamma))$, we get that Algorithm 2 has β-additive regret, completing the proof of Theorem 1.

D Analysis of FTRL (Proofs of Section 4)

In this chapter of the appendix we present all the omitted proofs from Section 4 concerning our analysis of the *Follow the Regularized Leader* (FTRL) algorithm (Algorithm 3). To avoid repetition, from now on we fix an arbitrary HST $\mathcal{T}$ and use $\mathcal{FP}(\mathcal{T})$ to denote the set of all fractional placements of k facilities on the leaves of $\mathcal{T}$. We use $n = |L(\mathcal{T})|$ to denote the number of leaves of $\mathcal{T}$, $h = h(\mathcal{T})$ to denote its height and $D = diam(\mathcal{T})$ to denote its diameter. Since $\mathcal{T}$ is an HST, we know that its diameter D, i.e. the maximum distance between any two leaves, is precisely $D = 2 \cdot (2^h - 1)$.

To ease notation, let $w_v = 2^{\mathrm{lev}(v)}$. For convenience, we remind the reader that our regularizer function $R_{\mathcal{T}} : \mathcal{FP}(\mathcal{T}) \mapsto \mathbb{R}$ is defined as

$$R_{\mathcal{T}}(y) = \sum_{v \neq r} w_v \cdot (y_v + \delta_v) \cdot \ln\left(\frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}\right)$$

where $\delta_v = k \cdot |L(\mathcal{T}) \cap T(v)|/|L(\mathcal{T})|$ is the percentage of leaves that lie on the sub-tree rooted at vertex v multiplied by k and $p(v)$ is the parent of node v. Also, recall that for any $y \in \mathcal{FP}(\mathcal{T})$ we have defined the norm

$$||y||_{\mathcal{T}} = \gamma \cdot \sum_{v \in V(\mathcal{T})} w_v |y_v|$$

Roadmap. In Section D.1 we prove Lemma 1, namely the strong convexity of $R_{\mathcal{T}}$ with respect to $||\cdot||_{\mathcal{T}}$. Then, in Section D.2 we bound the moving cost of FTRL, proving Lemma 2. Next, in Section D.3 we bound the connection cost cost of FTRL, proving Lemma 3. Finally, in Section D.4 we account for approximation errors in the computation of the regularized leader, proving Lemma 4.

D.1 Strong Convexity (Proof of Lemma 1)

The objective of this section is to prove Lemma 1, specifically that for any fractional facility placements $y, y' \in \mathcal{FP}(\mathcal{T})$ it holds that

$$R_{\mathcal{T}}(y') \geq R_{\mathcal{T}}(y) + \langle \nabla R_{\mathcal{T}}(y), y' - y \rangle + \alpha ||y - y'||^2_{\mathcal{T}}$$

where $\alpha = (8kD\gamma^2)^{-1}$.

We begin by computing the gradient of $R_{\mathcal{T}}$ on any fractional facility placement $y \in \mathcal{FP}(\mathcal{T})$.

Claim 3. *The partial derivatives of $R_{\mathcal{T}}$ on any point $y \in \mathcal{FP}(\mathcal{T})$ are given by*

$$\frac{\partial R_{\mathcal{T}}(y)}{\partial y_v} = \begin{cases} -\frac{w_v}{2} & \text{for } v = r \\ w_v \cdot \ln\left(\frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}\right) + w_v & \text{for } v \in L(\mathcal{T}) \\ w_v \cdot \ln\left(\frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}\right) + \frac{w_v}{2} & \text{for } v \notin L(\mathcal{T}) \cup \{r\} \end{cases}$$

Proof. Clearly, $R_{\mathcal{T}}$ is well-defined and differentiable on $\mathcal{FP}(\mathcal{T})$. For any $v \neq r$, we compute the partial derivatives of $R_{\mathcal{T}}(y)$ to obtain

$$\frac{\partial R_{\mathcal{T}}(y)}{\partial y_v} = w_v \cdot \ln\left(\frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}\right) + w_v - \sum_{v \in \mathrm{cld}(u)} w_u \cdot \frac{y_u + \delta_u}{y_v + \delta_v}$$

Since $y \in \mathcal{FP}(\mathcal{T})$, we know $y_v = \sum_{u \in \mathrm{cld}(v)} y_u$ and by definition, $\delta_v = \sum_{u \in \mathrm{cld}(v)} \delta_u$. Finally, recall that $w_u = w_v/2$ for any $u \in \mathrm{cld}(v)$. By plugging everything in we get

$$\frac{\partial R_{\mathcal{T}}(y)}{\partial y_v} = w_v \cdot \ln\left(\frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}\right) + w_v - \frac{w_v}{2} \cdot \mathbb{1}[v \notin L(\mathcal{T})]$$

for any $v \neq r$. For the root vertex, using similar arguments we get

$$\frac{\partial R_{\mathcal{T}}(y)}{\partial y_r} = -\frac{w_r}{2}$$

□

Now that we have calculated the gradient of $R_{\mathcal{T}}$, we can substitute it into the definition of strong convexity. Specifically, by Claim 3, Lemma 1 states that

$$\sum_{v\neq r} w_v \cdot (y'_v + \delta_v) \cdot \ln\left(\frac{\frac{y'_v+\delta_v}{y'_{p(v)}+\delta_{p(v)}}}{\frac{y_v+\delta_v}{y_{p(v)}+\delta_{p(v)}}}\right) \geq \frac{1}{8kD\gamma^2} \cdot ||y'-y||^2_{\mathcal{T}} \tag{1}$$

To ease the presentation, we define quantities

$$f(y',y) = \sum_{v\neq r} w_v \cdot (y'_v + \delta_v) \cdot \ln\left(\frac{\frac{y'_v+\delta_v}{y'_{p(v)}+\delta_{p(v)}}}{\frac{y_v+\delta_v}{y_{p(v)}+\delta_{p(v)}}}\right)$$

and

$$h(y',y) = \sum_{v\neq r} w_v \cdot (y_{p(v)} + \delta_{p(v)}) \cdot |\frac{y'_v+\delta_v}{y'_{p(v)}+\delta_{p(v)}} - \frac{y_v+\delta_v}{y_{p(v)}+\delta_{p(v)}}|$$

We will prove that $f(y',y) \geq (1/2kD)\cdot h^2(y',y)$ and that $h(y',y) \geq (1/2\gamma)\cdot||y'-y||_{\mathcal{T}}$ in Claims 4 and 5 respectively. Combining these claims, equation (1) clearly holds, completing the proof of Lemma 1.

Claim 4. *For any $y, y' \in \mathcal{FP}(\mathcal{T})$, it holds that $f(y',y) \geq \frac{1}{2kD}\cdot(h(y',y))^2$.*

Proof. We begin by establishing some notation. For any $v \neq r$, let

$$\mu'_v = w_v \cdot (y'_{p(v)} + \delta_{p(v)}) \cdot \frac{y'_v+\delta_v}{y'_{p(v)}+\delta_{p(v)}}$$

and

$$\mu_v = w_v \cdot (y'_{p(v)} + \delta_{p(v)}) \cdot \frac{y_v+\delta_v}{y_{p(v)}+\delta_{p(v)}}.$$

Then, we have that

$$\begin{aligned} f(y',y) &= \sum_{v\neq r} \mu'_v \cdot \ln\left(\frac{\mu'_v}{\mu_v}\right) \\ &= \sum_{v\in I} \mu'_v \cdot \ln\left(\frac{\mu'_v}{\mu_v}\right) + \sum_{v\in I'} \mu'_v \cdot \ln\left(\frac{\mu'_v}{\mu_v}\right) \end{aligned}$$

where $I = \{v \neq r : \mu'_v \geq \mu_v\}$ and $I' = \{v \neq r : \mu'_v < \mu_v\}$. By applying the log-sum inequality in both of these terms, we obtain

$$f(y',y) \geq (\sum_{v\in I}\mu'_v)\cdot\ln\left(\frac{\sum_{v\in I}\mu'_v}{\sum_{v\in I}\mu_v}\right) + (\sum_{v\in I'}\mu'_v)\cdot\ln\left(\frac{\sum_{v\in I'}\mu'_v}{\sum_{v\in I'}\mu_v}\right)$$

Next, observe that

$$\sum_{v\neq r}\mu'_v = \sum_{v\neq r} w_v\cdot(y'_v+\delta_v) = 2k\cdot(2^h-1) = k\cdot D$$

and also

$$\begin{aligned} \sum_{v\neq r}\mu_v &= \sum_{v\neq r} w_v \cdot (y'_{p(v)}+\delta_{p(v)})\cdot\frac{y_v+\delta_v}{y_{p(v)}+\delta_{p(v)}} \\ &= \sum_{v\notin L(\mathcal{T})}\left(\frac{w_v}{2}\cdot(y'_v+\delta_v)\cdot\sum_{u\in cld(v)}\frac{y_u+\delta_u}{y_v+\delta_v}\right) \\ &= \sum_{v\notin L(\mathcal{T})}\frac{w_v}{2}\cdot(y'_v+\delta_v) \\ &= \frac{1}{2}\cdot 2k\cdot(2^{h+1}-2) \\ &= k\cdot D. \end{aligned}$$

Let $B' = \sum_{v \in I} \mu'_v$ and $B = \sum_{v \in I} \mu_v$. Then, we have shown that

$$f(y', y) \geq B' \cdot \ln\left(\frac{B'}{B}\right) + (kD - B') \cdot \ln\left(\frac{kD - B'}{kD - B}\right) \tag{2}$$

Our next step is to apply Pinsker's inequality to the above expression. Pinsker's inequality states that for any $p, q \in (0, 1)$, it holds that

$$p \cdot \ln\left(\frac{p}{q}\right) + (1 - p) \cdot \ln\left(\frac{1-p}{1-q}\right) \geq 2 \cdot (p - q)^2$$

Since $B \leq kD$ and $B' \leq kD$, we can scale everything in inequality 2 and apply Pinsker's inequality to obtain

$$f(y', y) \geq \frac{2}{kD} \cdot (B - B')^2 \tag{3}$$

To complete the proof, we substitute

$$\begin{aligned}
B' - B &= \sum_{v \in I} (\mu'_v - \mu_v) \\
&= \sum_{v \in I} w_v \cdot (y'_{p(v)} + \delta_{p(v)}) \cdot (\frac{y'_v + \delta_v}{y'_{p(v)} + \delta_{p(v)}} - \frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}) \\
&= \sum_{v \notin L(\mathcal{T})} \frac{w_v}{2} \cdot (y'_v + \delta_v) \cdot \sum_{u \in \mathrm{cld}(v) \cap I} (\frac{y'_u + \delta_u}{y'_v + \delta_v} - \frac{y_u + \delta_u}{y_v + \delta_v}) \\
&= \frac{1}{2} \cdot \sum_{v \notin L(\mathcal{T})} \frac{w_v}{2} \cdot (y'_v + \delta_v) \cdot \sum_{u \in \mathrm{cld}(u)} |\frac{y'_u + \delta_u}{y'_v + \delta_v} - \frac{y_u + \delta_u}{y_v + \delta_v}|
\end{aligned}$$

where the last equality follows from the fact that the ratio in the inner sum always sum to 1, and thus by only summing over the ones with positive difference we get half of the total sum of absolute differences. By swapping the summation order once again, we get

$$\begin{aligned}
B' - B &= \frac{1}{2} \cdot \sum_{v \neq r} w_v \cdot (y'_{p(v)} + \delta_{p(v)}) \cdot |\frac{y'_v + \delta_v}{y'_{p(v)} + \delta_{p(v)}} - \frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}| \\
&= \frac{1}{2} \cdot h(y', y)
\end{aligned}$$

and from inequality (3) we finally get

$$f(y', y) \geq \frac{1}{2kD} \cdot (h(y', y))^2$$

as desired. $\square$

Claim 5. *For any $y, y' \in \mathcal{FP}(\mathcal{T})$, it holds that $||y' - y||_{\mathcal{T}} \leq 2\gamma \cdot h(y', y)$.*

Proof. To prove the claim, we first need to establish some extra notation. For any $y \in \mathcal{FP}(\mathcal{T})$ and $v \neq r$, let

$$\lambda_v(y) := \frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}$$

Furthermore, for any vertex v and any integer $i \in [0, h - lev(v)]$, we use $p(v, i)$ to denote the i-th ancestor of v on $\mathcal{T}$, for example $p(v, 0) = v$, $p(v, 1) = p(v)$ and $p(v, h - lev(v)) = r$.

Recall that by definition, $y_r = \delta_r = k$. Thus, if we telescope these terms and let $m_v = h - lev(v) - 1$, we clearly have that

$$y_v + \delta_v = 2k \cdot \Pi_{i=0}^{m_v} \lambda_{p(v,i)}(y)$$

which implies

$$\begin{aligned} y'_v - y_v &= 2k \cdot \Pi_{i=0}^{m_v} \lambda_{p(v,i)}(y') - 2k \cdot \Pi_{i=0}^{m_v} \lambda_{p(v,i)}(y) \\ &= 2k \cdot \sum_{i=0}^{m_v} \lambda_{p(v,0)}(y') \cdot \ldots \cdot (\lambda_{p(v,i)}(y') - \lambda_{p(v,i)}(y)) \cdot \ldots \cdot \lambda_{p(v,m_v)}(y) \\ &= 2k \cdot \sum_{i=0}^{m_v} \frac{y'_v + \delta_v}{y'_{p(v,i)} + \delta_{p(v,i)}} \cdot (\lambda_{p(v,i)}(y') - \lambda_{p(v,i)}(y)) \cdot \frac{y_{p(v,i+1)} + \delta_{p(v,i+1)}}{2k} \\ &= (y'_v + \delta_v) \cdot \sum_{i=0}^{m_v} \frac{y_{p(v,i+1)} + \delta_{p(v,i+1)}}{y'_{p(v,i)} + \delta_{p(v,i)}} \cdot (\lambda_{p(v,i)}(y') - \lambda_{p(v,i)}(y)) \end{aligned}$$

and from the triangular inequality

$$|y'_v - y_v| \leq (y'_v + \delta_v) \cdot \sum_{i=0}^{m_v} \frac{y_{p(v,i+1)} + \delta_{p(v,i+1)}}{y'_{p(v,i)} + \delta_{p(v,i)}} \cdot |\lambda_{p(v,i)}(y') - \lambda_{p(v,i)}(y)| \tag{4}$$

Plugging inequality (4) into the definition of norm $||\cdot||_{\mathcal{T}}$, we get

$$||y' - y||_{\mathcal{T}} \leq \gamma \cdot \sum_{v \neq r} w_v \cdot (y'_v + \delta_v) \cdot \left(\sum_{i=0}^{m_v} \frac{y_{p(v,i+1)} + \delta_{p(v,i+1)}}{y'_{p(v,i)} + \delta_{p(v,i)}} \cdot |\lambda_{p(v,i)}(y') - \lambda_{p(v,i)}(y)| \right)$$

and by carefully exchanging the summation order, we obtain

$$||y' - y||_{\mathcal{T}} \leq \gamma \cdot \sum_{v \neq r} \frac{y_{p(v)} + \delta_{p(v)}}{y'_v + \delta_v} \cdot |\lambda_v(y') - \lambda_v(y)| \cdot \left(\sum_{u \in T(v)} w_u (y'_u + \delta_u) \right)$$

Finally, observe that $\sum_{u \in T(v)} w_u y'_u \leq 2 w_v y'_v$. To see this, fix the sub-tree $T(v)$ rooted at vertex v and recall that since $y' \in \mathcal{FP}(\mathcal{T})$, the total amount of facilities at each level is y'_v. Furthermore, the weights w_v decrease by a factor of 2 at every level. Using the same arguments, we obtain $\sum_{u \in T(v)} w_u \delta_u \leq 2 w_v \delta_v$. Combining everything, we finally get

$$||y' - y||_{\mathcal{T}} \leq 2\gamma \cdot \sum_{v \neq r} w_v \cdot (y_{p(v)} + \delta_{p(v)}) \cdot |\lambda_v(y') - \lambda_v(y)|$$

or equivalently, $||y' - y||_{\mathcal{T}} \leq 2\gamma \cdot h(y', y)$. □

D.2 Bounding the Moving Cost (Proof of Lemma 2)

In this section we will upper bound the moving cost of FTRL by its connection cost. Fix any sequence of client requests $R_1, R_2, \ldots, R_T \subseteq L(\mathcal{T})$. Recall that at each step t, FTRL selects a fractional facility placement y^t given by

$$y^t = \operatorname*{arg\,min}_{y \in \mathcal{FP}(\mathcal{T})} \Phi_t(y)$$

where $\Phi_t(y) = \sum_{s=1}^{t-1} f_{R_s}(y) + \frac{1}{\eta} \cdot R_{\mathcal{T}}(y)$ is the objective that FTRL minimizes over at step t for $\eta = (\gamma \cdot \sqrt{nT})^{-1}$. In this section, we prove Lemma 2, by arguing that

$$\sum_{t=2}^{T} ||y^t - y^{t-1}||_{\mathcal{T}} \leq \frac{1}{2} \cdot \sum_{t=1}^{T} f_{R_t}(y^t) + \frac{\eta}{2\alpha} \cdot T$$

since the proof follows easily by the definitions of η and α.

From Lemma 1 we already know that $R_{\mathcal{T}}$ is α-strongly convex with respect to $||\cdot||_{\mathcal{T}}$ for $\alpha = (8kD\gamma^2)^{-1}$. Furthermore, by definition the fractional connection cost

$$f_R(y) = \sum_{j \in R} \sum_{v \in P(j,r)} 2^{lev(v)+1} \cdot \max(0, 1 - y_v)$$

is clearly convex for any client request $R \subseteq L(\mathcal{T})$. Thus, it is straight-forward to argue that at any step t, the FTRL objective Φ_t is $\frac{\alpha}{\eta}$-strongly convex with respect to $|| \cdot ||_{\mathcal{T}}$. Unfortunately, $f_R(y)$ is not differentiable on $\mathcal{FP}(\mathcal{T})$, but its sub-gradients are well-defined on any $y \in \mathcal{FP}(\mathcal{T})$. Thus, the strong convexity of Φ_t provides us with the following guarantee:

Claim 6. *Fix any pair of fractional facility placements $y, y' \in \mathcal{FP}(\mathcal{T})$ and any time step $t \in [T]$. Let $g_t \in \partial \Phi_t(y)$ be any sub-gradient of Φ_t at y. Then, it holds that*

$$\Phi_t(y') \geq \Phi_t(y) + \langle g_t, y' - y \rangle + \frac{\alpha}{\eta} \cdot ||y - y'||^2_{\mathcal{T}}$$

Furthermore, since by definition y^t is the (unique) minimizer of Φ_t, the first order optimality conditions on Φ_t imply that there exists some $g^*_t \in \partial \Phi_t(y^t)$ such that $\langle g^*_t, y - y^t \rangle \geq 0$ for any $y \in \mathcal{FP}(\mathcal{T})$. Claim 6 for $y = y^t$, $y' = y^{t-1}$ and $g_t = g^*_t$ gives us

$$\Phi_t(y^{t-1}) \geq \Phi_t(y^t) + \frac{\alpha}{\eta} \cdot ||y^t - y^{t-1}||^2_{\mathcal{T}}$$

Thus, we have

$$\begin{aligned} ||y^t - y^{t-1}||^2_{\mathcal{T}} &\leq \frac{\eta}{\alpha} \cdot \left(\Phi_t(y^{t-1}) - \Phi_t(y^t) \right) \\ &= \frac{\eta}{\alpha} \cdot \left(\Phi_{t-1}(y^{t-1}) + f_{R_{t-1}}(y^{t-1}) - \Phi_{t-1}(y^t) - f_{R_{t-1}}(y^t) \right) \\ &\leq \frac{\eta}{\alpha} \cdot \left(f_{R_{t-1}}(y^{t-1}) - f_{R_{t-1}}(y^t) \right) \end{aligned}$$

where for the equality we used the fact that $\Phi_t(y) = \Phi_{t-1}(y) + f_{R_{t-1}}(y)$ and for the second inequality we used the fact that y^{t-1} is by definition the minimizer of Φ_{t-1}. Finally, since $f_R(y) \geq 0$ for any client request $R \subseteq L(\mathcal{T})$, we have

$$\begin{aligned} ||y^t - y^{t-1}||_{\mathcal{T}} &\leq \sqrt{\frac{\eta}{\alpha} \cdot f_{R_{t-1}}(y^{t-1})} \\ &\leq \frac{\eta}{2\alpha} + \frac{1}{2} \cdot f_{R_{t-1}}(y^{t-1}) \end{aligned}$$

where the last inequality follows from the *Arithmetic Mean - Geometric Mean* inequality. Summing over all t completes the proof of Lemma 2.

D.3 Bounding the Connection Cost (Proof of Lemma 3)

In this section we will upper bound the connection cost of FTRL by the connection cost of the optimal fractional facility placement in hindsight. This is a standard analysis found in many textbooks, and we present it just for the sake of completeness.

Fix any sequence of client requests $R_1, R_2, \ldots, R_T \subseteq L(\mathcal{T})$. Recall that at each step t, FTRL selects a fractional facility placement y^t given by

$$y^t = \arg\min_{y \in \mathcal{FP}(\mathcal{T})} \Phi_t(y)$$

where $\Phi_t(y) = \sum_{s=1}^{t-1} f_{R_s}(y) + \frac{1}{\eta} \cdot R_{\mathcal{T}}(y)$ is the objective that FTRL minimizes over at step t for $\eta = (\gamma \cdot \sqrt{nT})^{-1}$. Let y^* be the optimal facility placement in hindsight, i.e.

$$y^* = \arg\min_{y \in \mathcal{FP}(\mathcal{T})} \sum_{t=1}^{T} f_{R_t}(y)$$

In this section we prove Lemma 3,by arguing that

$$\sum_{t=1}^{T} f_{R_t}(y^t) \leq \sum_{t=1}^{T} f_{R_t}(y^*) + \frac{knD}{\eta} + 32kn^2 D\eta \cdot T$$

and then the proof follows easily by definition of η.

In the standard analysis of FTRL, the following quantities are of special interest as they appear in the final regret guarantees of the algorithm:

- Let $\operatorname{diam}(R_{\mathcal{T}}) := \max_{y,y' \in \mathcal{FP}(\mathcal{T})} |R_{\mathcal{T}}(y) - R_{\mathcal{T}}(y')|$ be the diameter of the regularizer.
- Let G_f be an upper bound on the dual norm of the sub-gradient of the fractional connection cost for any client request, i.e. for any $R \subseteq L(\mathcal{T})$ and any $y \in \mathcal{FP}(\mathcal{T})$, there exists some sub-gradient $g \in \partial f_R(y)$ such that $||g||_{\mathcal{T}}^* \leq G_f$. Here, $||\cdot||_{\mathcal{T}}^*$ denotes the dual norm of $||\cdot||_{\mathcal{T}}$.

We begin by presenting the standard analysis of FTRL and deriving an expression for the regret guarantee that depends on the above quantities. Recall that at any step t, the FTRL objective Φ_t doesn't include f_{R_t} since the client request R_t is not revealed to the algorithm at the time of decision. We begin by bounding the connection cost of a theoretical algorithm that has access to this information and thus at time t can pick facility placement y^{t+1}.

Claim 7. *The output of* FTRL *satisfies*

$$\sum_{t=1}^{T} f_{R_t}(y^{t+1}) \leq \sum_{t=1}^{T} f_{R_t}(y^*) + \frac{\operatorname{diam}(R_{\mathcal{T}})}{\eta}$$

Proof. We have

$$\begin{aligned}\Phi_t(y^t) &= \Phi_{t-1}(y^t) + f_{R_{t-1}}(y^t)\\ &\geq \Phi_{t-1}(y^{t-1}) + f_{R_{t-1}}(y^t)\end{aligned}$$

where the equality holds by definition of Φ_t and the inequality holds from the optimality of y^{t-1} on Φ_{t-1}. Similarly, we obtain

$$\Phi_{t-1}(y^{t-1}) \geq \Phi_{t-2}(y^{t-2}) + f_{R_{t-2}}(y^{t-1})$$

If we keep applying this rule, we finally get that

$$\Phi_t(y^t) \geq \sum_{s=1}^{t-1} f_{R_s}(y^{s+1}) + \Phi_1(y^1)$$

Furthermore, we have $\Phi_1(y^1) = R_{\mathcal{T}}(y_1)/\eta$ and $\Phi_t(y^*) \geq \Phi_t(y^t)$ for all t. Thus, we get

$$\Phi_{T+1}(y^*) \geq \sum_{t=1}^{T} f_{R_t}(y^{t+1}) + \frac{1}{\eta} \cdot R_{\mathcal{T}}(y^1)$$

or equivalently (by substituting Φ_{T+1}'s definition) we have

$$\sum_{t=1}^{T} f_{R_t}(y^{t+1}) \leq \sum_{t=1}^{T} f_{R_t}(y^*) + \frac{R_{\mathcal{T}}(y^*) - R_{\mathcal{T}}(y^1)}{\eta}$$

The claim follows from the definition of $\operatorname{diam}(R_{\mathcal{T}})$. □

Next, we proceed by bounding the increase in the connection cost that we suffer by choosing y^t instead of y^{t+1} at time t.

Claim 8. *For any $t \geq 0$, it holds that $f_{R_t}(y^t) \leq f_{R_t}(y^{t+1}) + \eta G_f^2/\alpha$.*

Proof. For any client request $R \subseteq L(\mathcal{T})$, the fractional connection cost function $f_R(y)$ is clearly convex and its sub-gradients are well-defined on $\mathcal{FP}(\mathcal{T})$. By definition of G_f, we know that there exists some sub-gradient $g \in \partial f_{R_t}(y^t)$ such that $||g||_{\mathcal{T}}^* \leq G_f$. Using this sub-gradient, we get

$$\begin{aligned}f_{R_t}(y^t) &\leq f_{R_t}(y^{t+1}) + \langle g, y^t - y^{t+1}\rangle\\ &\leq f_{R_t}(y^{t+1}) + ||g||_{\mathcal{T}}^* \cdot ||y^t - y^{t+1}||_{\mathcal{T}}\\ &\leq f_{R_t}(y^{t+1}) + G_f \cdot ||y^t - y^{t+1}||_{\mathcal{T}}\end{aligned}$$

where the first inequality is derived from the convexity of the fractional connection cost, the second inequality is an application of Holder's inequality and the third inequality is from G_f's definition.

As we have already argued in section D.2, we know that for any step t, the FTRL objective Φ_t is α/η-strongly convex with respect to $||\cdot||_\mathcal{T}$. Using the definition of strong convexity, this implies that

$$\Phi_{t+1}(y^t) \geq \Phi_{t+1}(y^{t+1}) + \langle g, y^t - y^{t+1}\rangle + \frac{\alpha}{\eta}\cdot ||y^t - y^{t+1}||_\mathcal{T}^2$$

for any sub-gradient $g \in \partial\Phi_{t+1}(y^{t+1})$. Furthermore, since y^{t+1} is the minimizer of Φ_{t+1}, we know from the first order optimality conditions that we can select $g \in \partial\Phi_{t+1}(y^{t+1})$ such that $\langle g, y - y^{t+1}\rangle \geq 0$ for any $y \in \mathcal{FP}(\mathcal{T})$. Using such a sub-gradient, we get

$$\begin{aligned}
||y^t - y^{t+1}||_\mathcal{T}^2 &\leq \frac{\eta}{\alpha}\cdot\left(\Phi_{t+1}(y^t) - \Phi_{t+1}(y^{t+1})\right)\\
&= \frac{\eta}{\alpha}\cdot\left(\Phi_t(y^t) + f_{R_t}(y^t) - \Phi_t(y^{t+1}) - f_{R_t}(y^{t+1})\right)\\
&\leq \frac{\eta}{\alpha}\cdot\left(f_{R_t}(y^t) - f_{R_t}(y^{t+1})\right)
\end{aligned}$$

where we just expanded Φ_{t+1}'s definition and used the fact that y^t is the minimizer of Φ_t.

Combining everything, we finally obtain

$$f_{R_t}(y^t) - f_{R_t}(y^{t+1}) \leq G_f \cdot \sqrt{\frac{\eta}{\alpha}\cdot(f_{R_t}(y^t) - f_{R_t}(y^{t+1}))}$$

and the claim follows. □

We complete the analysis of FTRL by combining Claims 7 and 8 in order to obtain the following regret guarantee:

Claim 9. *The output of* FTRL *satisfies*

$$\sum_{t=1}^{T} f_{R_t}(y^t) \leq \sum_{t=1}^{T} f_{R_t}(y^*) + \frac{diam(R_\mathcal{T})}{\eta} + \frac{\eta G_f^2}{\alpha}\cdot T$$

It remains to substitute the specific values of the parameters that appear in the regret guarantee. We have already proven in section D.1 that $R_\mathcal{T}$ is α-strongly convex with respect to $||\cdot||_\mathcal{T}$ for $\alpha = (8kD\gamma^2)^{-1}$. Next, we provide an upper bound for the diameter of the regularizer.

Claim 10. *It holds that* $\mathrm{diam}(R_\mathcal{T}) \leq knD$.

Proof. Fix any $y \in \mathcal{FP}(\mathcal{T})$. Be definition, we know that $y_v \leq y_{p(v)}$ and $\delta_v \leq \delta_{p(v)}$ for any $v \neq r$. Thus, the expressions inside the logarithms of the regularizer are always at most 1, which implies that $R_\mathcal{T}(y) \leq 0$. Furthermore, for any $\alpha, \beta > 0$ it holds that $\alpha - \beta \leq \alpha \cdot \ln(\alpha/\beta)$. Using this inequality, we get that

$$R_\mathcal{T}(y) \geq \sum_{v\neq r} w_v \cdot (y_v + \delta_v - y_{p(v)} - \delta_{p(v)})$$

Fix any level $l \in [0, h-1]$ and let $V_l = \{v \in V(\mathcal{T}) : lev(v) = l\}$ denote the set of vertices of the HST at level l. Since $y \in \mathcal{FP}(\mathcal{T})$, we know that $\sum_{v\in V_l} y_v = k$, and by definition of δ's we know that $\sum_{v\in V_l}\delta_v = k$ as well. Furthermore, we know that $\sum_{v\in V_l} y_{p(v)} \leq n\cdot\sum_{v\in V_{l+1}} y_v = n\cdot k$ since any vertex v can have at most n (i.e. the total number of leaves) children. Using the same argument, we have $\sum_{v \in V_l} \delta_{p(v)} \leq n \cdot \sum_{v \in V_{l+1}} y_v = n \cdot k$. Thus, combining everything we obtain

$$\begin{aligned} R_{\mathcal{T}}(y) &\geq \sum_{v \neq r} w_v \cdot (y_v + \delta_v - y_{p(v)} - \delta_{p(v)}) \\ &= \sum_{l=0}^{h-1} \sum_{v \in V_l} 2^l \cdot (y_v + \delta_v - y_{p(v)} - \delta_{p(v)}) \\ &\geq \sum_{l=0}^{h-1} 2^l \cdot (2k - 2kn) \\ &= 2k(1-n)(2^h - 1) \\ &= k(1-n)D. \end{aligned}$$

which proves our claim. □

Finally, we only need to find an upper bound for G_f. We begin by computing a set of sub-gradients for the fractional connection cost function.

Claim 11. *Fix any client request $R \subseteq L(\mathcal{T})$ and any $y \in \mathcal{FP}(\mathcal{T})$. Define the vector $g^{R,y} \in \mathbb{R}^{|V(\mathcal{T})|}$ such that*

$$g_v^{R,y} = \begin{cases} 0 & \textit{if } y_v \geq 1 \\ -2^{lev(v)+1} \cdot |T(v) \cap R| & \textit{if } y_v < 1 \end{cases}$$

Then, $g^{R,y} \in \partial f_R(y)$, i.e. $g^{R,y}$ is a sub-gradient of f_R on point y.

Proof. Fix any client request $R \subseteq L(\mathcal{T})$. By definition of the fractional connection cost on facility placement $y \in \mathcal{FP}(\mathcal{T})$, we have

$$f_R(y) = \sum_{j \in R} \sum_{v \in P(j,r)} 2^{lev(v)+1} \cdot \max(0, 1 - y_v)$$

where $P(j, r)$ denotes the unique path from leaf $j \in L(\mathcal{T})$ to the root r. This is clearly a convex function on $\mathcal{FP}(\mathcal{T})$ and thus the sub-gradients of f_R are well-defined. Fix any $v \in V(\mathcal{T})$. We distinguish between two cases.

- If $y_v < 1$, then the partial derivative of $f_R(y)$ is well-defined and given by

  $$\frac{\partial f_R(y)}{\partial y_v} = -2^{lev(v)+1} \cdot |T(v) \cap R|$$

  where $T(v)$ is the set of vertices on the sub-tree rooted at vertex v.
- If $y_v \geq 1$, then clearly it doesn't contribute to $f_R(y)$. Using standard calculus, it is not hard to argue that in this case there exists a sub-gradient of $f_R(y)$ whose coordinate corresponding to v is 0. Thus, we have argued that that $g^{R,y}$ is a valid sub-gradient of f_R on point y.

□

Finally, we provide an upper bound on the dual-norm of the sub-gradients that we computed on Claim 11.

Claim 12. *For any $y \in \mathcal{FP}(\mathcal{T})$ and any $R \subseteq L(\mathcal{T})$, it holds that $||g^{R,v}||_{\mathcal{T}}^* \leq \frac{2n}{\gamma}$.*

Proof. Recall that we have defined the moving cost norm as

$$||y||_{\mathcal{T}} = \gamma \cdot \sum_{v \in V(\mathcal{T})} w_v \cdot y_v$$

which is basically a weighted l_1-norm with weights $\gamma \cdot w_v$. It is well-known that the dual of the l_1-norm is the l_∞ norm. Similarly, the dual of the weighted l_1-norm is a weighted l_∞ norm with inverse weights, i.e. $||\cdot||^* = l_\infty((\gamma w)^{-1})$. Thus, we have

$$||x||_{\mathcal{T}}^* = \max_v \frac{|x_v|}{\gamma \cdot w_v}$$

Using the calculation of the sub-gradients from Claim 11 and that $R \subseteq L(\mathcal{T})$ and thus $|R| \leq n$, we immediately get the claim. □

Claim 10 provides us with an expression for $\text{diam}(R_{\mathcal{T}})$ and Claim 12 provides us with an expression for G_f. Plugging everything in into Claim 9, we complete the proof of Lemma 3.

D.4 Incorporating approximation errors (Proof of Lemma 4)

Fix any sequence of client requests $R_1, R_2, \ldots, R_T \subseteq L(\mathcal{T})$. Recall that at each step t, FTRL selects a fractional facility placement y^t given by

$$y^t = \arg\min_{y \in \mathcal{FP}(\mathcal{T})} \Phi_t(y)$$

where $\Phi_t(y) = \sum_{s=1}^{t-1} f_{R_s}(y) + \frac{1}{\eta} \cdot R_{\mathcal{T}}(y)$ is the objective that FTRL minimizes over at step t for $\eta = (\gamma \cdot \sqrt{nT})^{-1}$.

Now, assume that instead of minimizing $\Phi_t(y)$ over $\mathcal{FP}(\mathcal{T})$ to compute y^t, we are only able to compute a fractional facility placement $z^t \in \mathcal{FP}(\mathcal{T})$ such that $\Phi_t(z^t) \leq \Phi_t(y^t) + \epsilon$ for some $\epsilon > 0$.

Claim 13. *For any step t, it holds that*

$$||z^t - y^t||_{\mathcal{T}} \leq \sqrt{\epsilon \cdot \frac{\eta}{\alpha}}$$

Proof. As we have already argued in section D.2, we know that for any step t, the FTRL objective Φ_t is α/η-strongly convex with respect to $||\cdot||_{\mathcal{T}}$. Combining this with the first order optimality condition for Φ_t on y_t, we get

$$\Phi_t(z^t) \geq \Phi_t(y^t) + \frac{\alpha}{\eta} \cdot ||z^t - y^t||_{\mathcal{T}}^2$$

which implies that

$$||z^t - y^t||_{\mathcal{T}} \leq \sqrt{\epsilon \cdot \frac{\eta}{\alpha}}$$

□

Using Claim 13, we can easily bound both the connection and the moving cost of the approximated FTRL solutions.

- For the connection cost, recall that the fractional connection cost function f_{R_t} at step t is convex, which implies that

  $$f_{R_t}(z^t) \leq f_{R_t}(y^t) + \langle g, z^t - y^t \rangle$$

  for some $g \in \partial f_{R_t}(z^t)$. Using Holder's inequality to upper bound the inner-product and using the upper bound of Claim 12 for the dual norm of the sub-gradients of f_{R_t}, we get that

  $$f_{R_t}(z^t) \leq f_{R_t}(y^t) + \frac{2n}{\gamma} \cdot ||z^t - y^t||_{\mathcal{T}}$$

  and finally from Claim 13 we get that

  $$f_{R_t}(z^t) \leq f_{R_t}(y^t) + \frac{2n}{\gamma} \cdot \sqrt{\epsilon \cdot \frac{\eta}{\alpha}}$$

- For the moving cost, recall it suffices to use the triangular inequality that $||\cdot||_{\mathcal{T}}$ (as a norm) satisfies:

$$\begin{aligned}||z^t - z^{t-1}||_{\mathcal{T}} &\leq ||z^t - y^t||_{\mathcal{T}} + ||y^t - y^{t-1}||_{\mathcal{T}} + ||y^{t-1} - z^{t-1}||_{\mathcal{T}} \\ &\leq ||y^t - y^{t-1}||_{\mathcal{T}} + 2 \cdot \sqrt{\epsilon \cdot \frac{\eta}{\alpha}}\end{aligned}$$

The proof of Lemma 4 follows easily by plugging in $\eta = (\gamma \cdot \sqrt{nT})^{-1}$, $\alpha = (8kD\gamma^2)^{-1}$ and $\epsilon = \mathcal{O}(1/\sqrt{T})$.

D.5 Implementation of Projected Mirror Descent

We conclude this section by considering the *Projected Mirror Descent* update step, namely

$$y' = \underset{y^* \in \mathcal{FP}(\mathcal{T})}{\arg\min} \; [\eta \cdot \langle c, y^* \rangle + \cdot D_{R_{\mathcal{T}}}(y^*, y)]$$

that takes as input a fractional facility placement $y \in \mathcal{FP}(\mathcal{T})$ and returns some other $y' \in \mathcal{FP}(\mathcal{T})$ that minimizes a linear cost under vector c plus the Bregman Divergence between the initial and the new point under regularizer $R_{\mathcal{T}}$. Here, $\eta > 0$ is a tuning parameter that balances the dynamics between the linear cost and the Bregman Divergence.

By letting c be the sub-gradient of the fractional connection cost over the observed sequence of clients, we can use this update step in order to approximate the FTRL objective; this is, in fact, the implementation we did for our experimental evaluation of Algorithm 2. In this section we will argue that the special structure of $R_{\mathcal{T}}$ allows us to compute the update step in linear (to the size of the HST) time.

By definition of the Bregman Divergence, we have

$$D_{R_{\mathcal{T}}}(x, y) = R_{\mathcal{T}}(x) - R_{\mathcal{T}}(y) - \langle \nabla R_{\mathcal{T}}(y), x - y \rangle$$

Substituting everything, we get that the update step of *Projected Mirror Descent* can be written as

$$y' = \underset{y^* \in \mathcal{FP}(\mathcal{T})}{\arg\min} \; F(y^*)$$

for

$$\begin{aligned}F(y^*) = \eta \cdot \sum_v c_v \cdot y_v^* &+ \sum_{v \neq r} w_v \cdot (y_v^* + \delta_v) \cdot \ln\left(\frac{y_v^* + \delta_v}{y_{p(v)}^* + \delta_{p(v)}}\right) \\ &- \sum_{v \neq r} w_v \cdot (y_v + \delta_v) \cdot \ln\left(\frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}\right) \\ &- \sum_{v \neq r} \left(w_v \cdot \ln\left(\frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}\right) + \frac{w_v}{2} + \frac{w_v}{2} \cdot \mathbb{1}[v \in L(\mathcal{T})]\right)(y_v^* - y_v) \\ &- \frac{w_r}{2}(y_r^* - y_r)\end{aligned}$$

It is always the case that we update y' from some $y \in \mathcal{FP}(\mathcal{T})$, so we can simplify the above expression to get

$$\begin{aligned}F(y^*) = \eta \cdot \sum_v c_v \cdot y_v^* + \sum_{v \neq r} w_v \cdot (y_v^* + \delta_v) \cdot \ln\left(\frac{\frac{y_v^* + \delta_v}{y_{p(v)}^* + \delta_{p(v)}}}{\frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}}}\right) \\ - \sum_v \frac{w_v}{2} \cdot (1 + \mathbb{1}[v \in L(\mathcal{T})]) \cdot (y_v^* - y_v)\end{aligned}$$

Recall that by definition, $\mathcal{FP}(T)$ is the polytope

$$\mathcal{FP}(\mathcal{T}) = \begin{cases} y \in \mathbb{R}^{|V(\mathcal{T})|} : & \begin{array}{ll} y_v = \sum_{u \in \mathrm{cld}(v)} y_u & v \notin L(\mathcal{T}) \\ y_v \in [0,1] & v \in L(\mathcal{T}) \\ y_r = k & \end{array} \end{cases}$$

Since our objective is to minimize function $F(\cdot)$ over $\mathcal{FP}(\mathcal{T})$, we can write down the KKT optimality conditions to obtain the following conditions about the minimizer y^*:

$$\frac{y_v^* + \delta_v}{y_{p(v)}^* + \delta_{p(v)}} = \frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}} \cdot \exp\left(\frac{1}{w_v}(\mu_{p(v)} - \mu_v - \eta c_v)\right)$$

where μ_v is the Lagrange multiplier for constraint $y_v = \sum_{u \in \mathrm{cld}(v)} y_u$ and $\mu_v = 0$ for $v \in L(\mathcal{T})$. To complete our computation of y^*, it remains to compute the Lagrange multipliers μ.

Since $y^* \in \mathcal{FP}(\mathcal{T})$, it is not hard to verify that for any $v \notin L(\mathcal{T})$ it holds

$$\sum_{u \in \mathrm{cld}(v)} \frac{y_u^* + \delta_u}{y_v^* + \delta_v} = 1$$

and using the KKT optimality condition, this implies that for any $v \notin L(\mathcal{T})$

$$\sum_{u \in \mathrm{cld}(v)} \frac{y_u + \delta_u}{y_v + \delta_v} \cdot \exp\left(\frac{1}{w_u}(\mu_v - \mu_u - \eta c_u)\right) = 1$$

or equivalently, since $w_v = 2w_u$ for all $u \in \mathrm{cld}(v)$,

$$\mu_v = -\frac{w_v}{2} \cdot \ln\left(\sum_{u \in \mathrm{cld}(v)} \frac{y_u + \delta_u}{y_v + \delta_v} \cdot \exp\left(-\frac{\mu_u + \eta c_u}{w_u}\right)\right)$$

Thus, starting from $\mu_v = 0$ on the leaves, this expression provides as a bottom-up algorithm to compute all the Lagrange multipliers μ. Using these multipliers and the KKT optimality conditions, we can then easily compute the ratios

$$\frac{y_v^* + \delta_v}{y_{p(v)}^* + \delta_{p(v)}} = \frac{y_v + \delta_v}{y_{p(v)} + \delta_{p(v)}} \cdot \exp\left(\frac{1}{w_v}(\mu_{p(v)} - \mu_v - \eta c_v)\right)$$

for all vertices $v \neq r$. Finally, we can start from the root vertex r, for which we know that $y_r^* = k$, and cascade these ratios downwards until we reach the leaves and we have compute all entries of y^*. Clearly, this is all done in linear time to the number of vertices.

Intuitively, this update step can be interpreted as an application of the Multiplicative Weights Update algorithm on every parent vertex v that decides how its mass should be split to its children. We repeat this process in a bottom-up manner, and then we simply start with k facilities on the root and begin splitting them based on these ratios while moving downwards.

E Analysis of Cut&Round (Proofs of Section 5)

In this chapter of the appendix we present all the omitted proofs from Section 5 concerning our online rounding scheme $\text{Cut\&Round}$. To avoid repetition, from now on we fix an arbitrary HST $\mathcal{T}$ and use $\mathcal{FP}(\mathcal{T})$ to denote the set of all fractional placements of k facilities on the leaves of $\mathcal{T}$.

Roadmap. In section E.1, we argue about the correctness of $\text{Cut\&Round}$; namely, we show that no matter the input, $\text{Cut\&Round}$ always returns a set of k-leaves of $\mathcal{T}$ where the facilities are placed. Then, in section E.2 we establish the main property of $\text{Cut\&Round}$ and prove Lemma 5. Finally, in section E.3 we analyze the expected connection cost of $\text{Cut\&Round}$'s output and prove Item 1 of Theorem 4 (Lemma 6) while in section E.4 we analyze the expected moving cost of $\text{Cut\&Round}$'s output and prove Item 2 of Theorem 4 (Lemma 7).

E.1 Correctness of Cut&Round

We begin by proving the correctness of $\text{Cut\&Round}$. Fix any $y \in \mathcal{FP}(\mathcal{T})$ and any set of thresholds $\alpha \in [0,1]^{|V(\mathcal{T})|}$. Let $F = \text{Cut\&Round}(\mathcal{T}, y, \alpha)$. In this section, we will prove that $|F| = k$, i.e. we will argue that $\text{Cut\&Round}$ always returns a set of k leaves at which facilities must be placed, as it is expected to. In order to show this, we will need to analyze the Y_v variables produced by $\text{Cut\&Round}$.

Claim 14. *For any leaf $v \in L(\mathcal{T})$, it holds that $Y_v \in \{0, 1\}$.*

Proof. Observe that for any $v \in V(\mathcal{T})$, sub-routine Alloc sets Y_v to either $\lfloor y_v \rfloor$ or $\lfloor y_v \rfloor + 1$. By definition of $\mathcal{FP}(\mathcal{T})$, we have $y_v \in [0, 1]$ for each leaf $v \in L(\mathcal{T})$. We distinguish between two different cases. If $y_v \in [0, 1)$, then clearly $Y_v \in \{0, 1\}$. If $y_v = 1$, then $\delta(y_v) = 0$ and thus Alloc will always set $Y_v = \lfloor y_v \rfloor = 1$. Thus, the claim holds for all leaves $v \in L(\mathcal{T})$. □

Claim 15. *Let $v \notin L(\mathcal{T})$ be any non-leaf vertex. Then, $Y_v = \sum_{u \in \text{cld}(v)} Y_u$.*

Proof. Fix any non-leaf vertex $v \notin L(\mathcal{T})$. We will analyze the inner loop of $\text{Cut\&Round}$ that iterates over v's children. Initially, $\text{Cut\&Round}$ sets $Y_{rem} = Y_v$ and $y_{rem} = y_v$. Then, we proceed to itteratively call Alloc, once per child vertex of v. Each time Alloc assigns some value Y_u to a child vertex $u \in \text{cld}(v)$, we update Y_{rem} to $Y_{rem} - Y_u$; thus, to prove our claim it suffices to argue that after we update the last child vertex, we have $Y_{rem} = 0$.

Since by definition of sub-routine Alloc we know that $Y_v \in \{\lfloor y_v \rfloor, \lfloor y_v \rfloor + 1\}$, we know that initially (before any child vertex is assigned a value Y_u) it holds that $Y_{rem} \in \{\lfloor y_{rem} \rfloor, \lfloor y_{rem} \rfloor + 1\}$. In fact, a simple case analysis over the decision tree of sub-routine Alloc suffices to see that this invariant holds not only at the beginning, but even after we begin assigning values to the child vertices and update Y_{rem} and y_{rem}.

Since $y \in \mathcal{FP}(\mathcal{T})$, we know that $y_v = \sum_{u \in \text{cld}(v)} y_u$ and thus $y_{rem} = y_u$ at the time we iterate over the last child vertex $u \in \text{cld}(v)$. Furthermore, from the above discussion we know that $Y_{rem} \in \{\lfloor y_u \rfloor, \lfloor y_u \rfloor + 1\}$. Since $\delta(y_u) = \delta(y_{rem})$, it is easy to verify that in any case Alloc sets $Y_u = Y_{rem}$ and thus after the last update we have $Y_{rem} = 0$, as desired. □

Proof of Correctness. Recall that by definition, the output of $\text{Cut\&Round}$ is $F = \{v \in L(\mathcal{T}) : Y_v = 1\}$. Since from Claim 14 we know that $Y_v \in \{0, 1\}$ for all $v \in L(\mathcal{T})$, this implies that $|F| = \sum_{v \in L(\mathcal{T})} Y_v$. We apply Claim 15 to the root vertex r, then again to each $u \in \text{cld}(r)$ and so on until we reach the leaves. This gives us that $Y_r = \sum_{v \in L(\mathcal{T})} Y_v$ and thus $|F| = Y_r$. Since by definition $\text{Cut\&Round}$ sets $Y_r = k$, we have proven that $|F| = k$ as desired.

E.2 Proof of Lemma 5 (Computing the Allocation Probabilities)

In this section, we formally prove the main property of algorithm $\mathrm{Cut\&Round}$, as stated in Lemma 5. Fix any fractional facility placement $y \in \mathcal{FP}(\mathcal{T})$ and let $\alpha_v \sim \mathrm{Unif}(0,1)$ be independent uniformly random thresholds for all $v \in V(\mathcal{T})$. Let $F = \mathrm{Cut\&Round}(\mathcal{T}, y, \alpha)$ be the output of algorithm $\mathrm{Cut\&Round}$ on this set of inputs. Recall that algorithm $\mathrm{Cut\&Round}$ sets the variables Y_v during its execution, for all $v \in V(\mathcal{T})$. As we have already discussed, Y_v is the total number of facilities in F on the leaves of the sub-tree rooted at v, i.e. $Y_v = |T(v) \cap F|$. We will prove that for any $v \in V(\mathcal{T})$, we have

$$Y_v = \begin{cases} \lfloor y_v \rfloor & \text{with probability } 1 - \delta(y_v) \\ \lfloor y_v \rfloor + 1 & \text{with probability } \delta(y_v) \end{cases}$$

We begin by writing down the following property for sub-routine Alloc:

Claim 16. *Fix any fractional facility placement $y \in \mathcal{FP}(\mathcal{T})$ and let $\alpha_v \sim \mathrm{Unif}(0,1)$ for all $v \in V(\mathcal{T})$. For any vertex $u \in V(\mathcal{T})$ of $\mathcal{T}$, let $Y_u = \mathrm{Alloc}(y_u, y_{rem}, Y_{rem}, \alpha_u)$ be the number of facilities assigned to the sub-tree of u by Line 8 of Algorithm $\mathrm{Cut\&Round}$ (Algorithm 4). Then,*

$$\mathbb{P}_\alpha\left[Y_u = \lfloor y_u \rfloor\right] = \begin{cases} 1 & \text{if } Y_{rem} = \lfloor y_{rem} \rfloor \text{and } \delta(y_u) \le \delta(y_{rem}) \\ \frac{1-\delta(y_u)}{1-\delta(y_{rem})} & \text{if } Y_{rem} = \lfloor y_{rem} \rfloor \text{and } \delta(y_u) > \delta(y_{rem}) \\ 0 & \text{if } Y_{rem} \neq \lfloor y_{rem} \rfloor \text{and } \delta(y_u) > \delta(y_{rem}) \\ \frac{\delta(y_{rem})-\delta(y_u)}{\delta(y_{rem})} & \text{if } Y_{rem} \neq \lfloor y_{rem} \rfloor \text{and } \delta(y_u) \le \delta(y_{rem}) \end{cases}$$

Proof. This claim is a direct consequence of sub-routine Alloc's description (Algorithm 5) and the fact that $\alpha_v \sim \mathrm{Unif}(0,1)$ for all $v \in V(\mathcal{T})$. ☐

Using this claim, we are now ready to prove Lemma 5.

Proof of Lemma 5. We prove the lemma via a top-down induction on the vertices of $\mathcal{T}$ (decreasing level order). For the root vertex, we know that since $y \in \mathcal{FP}(\mathcal{T})$ we have $y_r = k$ and also by definition of $\mathrm{Cut\&Round}$ we have $Y_r = k$. Thus, we get that $Y_r = y_r = \lfloor y_r \rfloor$ with probability $1-\delta(y_r) = 1$ and the claim holds. Now, fix any non-leaf vertex $v \notin L(\mathcal{T})$ and assume that $Y_v = \lfloor y_v \rfloor$ with probability $1 - \delta(y_v)$ and $Y_v = \lfloor y_v \rfloor + 1$ with probability $\delta(y_v)$. To complete our induction, we will now proceed to prove the claim for all the children vertices of v.

We begin by proving the claim for the first child of vertex v, and then we will show how the same arguments extend for all its children. Let $u \in \mathrm{cld}(v)$ be the *first* child vertex of v that $\mathrm{Cut\&Round}$ iterates over. Then, by definition of $\mathrm{Cut\&Round}$ we have that $Y_{rem} = Y_v$ and $y_{rem} = y_v$. Using the inductive hypothesis on v, this implies that $Y_{rem} = \lfloor y_{rem} \rfloor$ with probability $1 - \delta(y_{rem})$ and $Y_{rem} = \lfloor y_{rem} \rfloor + 1$ with probability $\delta(y_{rem})$. Conditioning on the value of Y_{rem}, we get

$$\begin{aligned} \mathbb{P}_\alpha\left[Y_u = \lfloor y_u \rfloor\right] = \;& \mathbb{P}_\alpha\left[Y_u = \lfloor y_u \rfloor \mid Y_{rem} = \lfloor y_{rem} \rfloor\right] \cdot (1 - \delta(y_{rem})) \\ & + \mathbb{P}_\alpha\left[Y_u = \lfloor y_u \rfloor \mid Y_{rem} = \lfloor y_{rem} \rfloor + 1\right] \cdot \delta(y_{rem}) \end{aligned}$$

We distinguish between two different cases based on whether $\delta(y_u) \le \delta(y_{rem})$ or $\delta(y_u) > \delta(y_{rem})$. In any case, we can use Claim 16 to substitute the conditional probabilities on the above expression and easily get that

$$\mathbb{P}_\alpha\left[Y_u = \lfloor y_u \rfloor\right] = 1 - \delta(y_u)$$

Thus, we have already proven the claim for the first child of v. However, to complete our induction, we need to prove the claim for all children of v and not just the first one. The only property that we used and holds specifically for the first child was that $Y_{rem} = \lfloor y_{rem} \rfloor$ with probability $1 - \delta(y_{rem})$ and $Y_{rem} = \lfloor y_{rem} \rfloor + 1$ with probability $\delta(y_{rem})$. Let Y'_{rem} and y'_{rem} be the updated remaining facilities after the value Y_u of the first child has been assigned. If we can prove that $Y'_{rem} = \lfloor y'_{rem} \rfloor$ with probability $1 - \delta(y'_{rem})$ and $Y'_{rem} = \lfloor y'_{rem} \rfloor + 1$ with probability $\delta(y'_{rem})$, then we can keep applying the same argument and inductively prove the claim for all the children of v.

By definition, we have that $Y'_{rem} = Y_{rem} - Y_u$ and $y'_{rem} = y_{rem} - y_u$. Once again, we distinguish between two different cases.

- Let $\delta(y_u) \leq \delta(y_{rem})$. In that case, we get that $\lfloor y'_{rem} \rfloor = \lfloor y_{rem} \rfloor - \lfloor y_u \rfloor$ and also that $\delta(y'_{rem}) = \delta(y_{rem}) - \delta(y_u)$. Since we know that $Y_{rem} \in \{\lfloor y_{rem} \rfloor, \lfloor y_{rem} \rfloor + 1\}$ and $Y_u \in \{\lfloor y_u \rfloor, \lfloor y_u \rfloor + 1\}$, this implies that

$$\begin{aligned}\mathbb{P}_\alpha[Y'_{rem} = \lfloor y'_{rem} \rfloor] = \mathbb{P}_\alpha[Y_{rem} = \lfloor y_{rem} \rfloor \;\cap\; Y_u = \lfloor y_u \rfloor] \\ + \mathbb{P}_\alpha[Y_{rem} = \lfloor y_{rem} \rfloor + 1 \;\cap\; Y_u = \lfloor y_u \rfloor + 1]\end{aligned}$$

  Using conditional probabilities and the inductive hypothesis on the distribution of Y_{rem}, we obtain

$$\begin{aligned}\mathbb{P}_\alpha\left[Y'_{rem} = \lfloor y'_{rem} \rfloor\right] = \mathbb{P}_\alpha\left[Y_u = \lfloor y_u \rfloor \mid Y_{rem} = \lfloor y_{rem} \rfloor\right] \cdot (1 - \delta(y_{rem})) \\ + \mathbb{P}_\alpha\left[Y_u = \lfloor y_u + 1 \rfloor \mid Y_{rem} = \lfloor y_{rem} \rfloor + 1\right] \cdot \delta(y_{rem})\end{aligned}$$

  Using Claim 16 to substitute the conditional probabilities, we finally get

$$\mathbb{P}_\alpha\left[Y'_{rem} = \lfloor y'_{rem} \rfloor\right] = 1 - \delta(y_{rem}) + \delta(y_u) = 1 - \delta(y'_{rem})$$

  as desired.

- Let $\delta(y_u) > \delta(y_{rem})$. In that case, we get that $\lfloor y'_{rem} \rfloor = \lfloor y_{rem} \rfloor - \lfloor y_u \rfloor - 1$ and also that $\delta(y'_{rem}) = 1 + \delta(y_{rem}) - \delta(y_u)$. Since we know that $Y_{rem} \in \{\lfloor y_{rem} \rfloor, \lfloor y_{rem} \rfloor + 1\}$ and $Y_u \in \{\lfloor y_u \rfloor, \lfloor y_u \rfloor + 1\}$, this implies that

$$\mathbb{P}_\alpha\left[Y'_{rem} = \lfloor y'_{rem} \rfloor\right] = \mathbb{P}_\alpha\left[Y_{rem} = \lfloor y_{rem} \rfloor \;\cap\; Y_u = \lfloor y_u \rfloor + 1\right]$$

  Using conditional probabilities and the inductive hypothesis on the distribution of Y_{rem}, we obtain

$$\mathbb{P}_\alpha\left[Y'_{rem} = \lfloor y'_{rem} \rfloor\right] = \mathbb{P}_\alpha\left[Y_u = \lfloor y_u \rfloor + 1 \mid Y_{rem} = \lfloor y_{rem} \rfloor\right] \cdot (1 - \delta(y_{rem}))$$

  Using Claim 16 to substitute the conditional probabilities, we finally get

$$\mathbb{P}_\alpha\left[Y'_{rem} = \lfloor y'_{rem} \rfloor\right] = \delta(y_u) - \delta(y_{rem}) = 1 - \delta(y'_{rem})$$

  as desired.

Thus, we have concluded the proof of Lemma 5.

E.3 Proof of Item 1 in Theorem 4 (Bounding the Expected Connection Cost)

Lemma 6. *Let* $F = \text{Cut\&Round}(y, \alpha)$ *where for all* $v \in V(\mathcal{T})$*,* $\alpha_v \sim \text{Unif}(0, 1)$ *independently. Then,*

$$\mathbb{E}_\alpha[C_R(F)] = f_R(y) \;\; \textit{for any } R \subseteq L(\mathcal{T})$$

Proof. Fix any $y \in \mathcal{FP}(\mathcal{T})$ and let $\alpha \in [0,1]^{|V(\mathcal{T})|}$ be a set of thresholds such that for each $v \in V(\mathcal{T})$, α_v is drawn independently at random from the uniform distribution, i.e. $\alpha_v \sim \text{Unif}(0, 1)$. Let $F = \text{Cut\&Round}(\mathcal{T}, y, \alpha)$. We will prove that for any set of clients $R \subseteq L(\mathcal{T})$, it holds that $\mathbb{E}_\alpha[C_R(F)] = f_R(y)$.

Recall that the Y_v variables set by Cut&Round denote the total number of facilities in F that are placed on the sub-tree rooted at vertex v, i.e. $Y_v = |F \cap T(v)|$. As argued in section E.1, we know that $Y \in \mathcal{FP}(\mathcal{T}) \cap \mathbb{N}$, i.e. Y is a valid integral facility placement. Thus, from Claim 1 of section C.1, we know that $C_R(F) = f_R(Y)$. This implies that by definition of the fractional connection cost under client request R, we have that

$$C_R(F) = \sum_{j \in R} \sum_{v \in P(j,r)} 2^{\text{lev}(v)+1} \cdot \max\left(0, 1 - Y_v\right)$$

Thus, we get

$$\begin{aligned}\mathbb{E}_\alpha[C_R(F)] &= \sum_{j \in R} \sum_{v \in P(j,r)} 2^{\text{lev}(v)+1} \cdot \mathbb{E}_\alpha[\max\left(0, 1 - Y_v\right)] \\ &= \sum_{j \in R} \sum_{v \in P(j,r)} 2^{\text{lev}(v)+1} \cdot \mathbb{P}_\alpha[Y_v = 0]\end{aligned}$$

where the first equality holds by linearity of expectation, and the second equality holds by the fact that $Y_v \in \mathbb{N}$ for all $v \in V(\mathcal{T})$. Since $Y_v \in \{\lfloor y_v \rfloor, \lfloor y_v \rfloor + 1\}$, we know that for any $v \in V(\mathcal{T})$, Y_v can be 0 only if $y_v \in [0, 1)$. Furthermore, from Lemma 5, we know that in the case of uniformly random thresholds, this happens with probability precisely $1 - y_v$. Combining these facts, we get $\mathbb{P}_\alpha[Y_v = 0] = \max(0, 1 - y_v)$ and thus

$$\begin{aligned}\mathbb{E}_\alpha[C_R(F)] &= \sum_{j \in R} \sum_{v \in P(j,r)} 2^{\mathrm{lev}(v)+1} \cdot \max(0, 1 - y_v) \\ &= f_R(y)\end{aligned}$$

concluding the proof of Lemma 6. □

E.4 Proof of Item 2 in Theorem 4 (Bounding the Expected Moving Cost)

Lemma 7. *Let $F = \mathrm{Round\&Cut}(y, \alpha)$ and also let $F' = \mathrm{Round\&Cut}(\mathcal{T}, y', \alpha)$ where $\alpha_v \sim \mathrm{Unif}(0, 1)$ for all $v \in V(\mathcal{T})$. Then,*

$$\gamma \cdot \mathbb{E}_\alpha \left[M_\mathcal{T}(F, F')\right] \le 4 \cdot ||y - y'||_\mathcal{T}$$

Proof. Fix any pair of fractional facility placements $y, y' \in \mathcal{FP}(\mathcal{T})$ and let corresponding outputs of $\mathrm{Cut\&Round}$ be denoted as $F = \mathrm{Cut\&Round}(\mathcal{T}, y, \alpha)$ and $F' = \mathrm{Cut\&Round}(\mathcal{T}, y', \alpha)$. Observe that the same set of (uniformly random) thresholds α_v is used in both cases, as this will play a crucial part in our analysis. To prove Lemma 7, we need to prove that

$$\gamma \cdot \mathbb{E}_\alpha \left[M_\mathcal{T}(F, F')\right] \le 4 \cdot ||y - y'||_\mathcal{T}$$

where the expectation is taken over the value of the uniformly random thresholds α_v.

The proof of Lemma 7 is technically involved, and thus we will break down our approach into smaller sections to ease the presentation. We begin by proving the Lemma in the special case where the transition from y to y' has a very simple structure, which we now proceed to define:

Definition 10. *We say that two fractional facility placements $y, y' \in \mathcal{FP}(\mathcal{T})$ are ϵ-neighboring if there are two leaves $s, t \in L(\mathcal{T})$ with least common ancestor $p \in V(\mathcal{T})$ such that the following hold:*

1. *$y'_v = y_v - \epsilon$ for all $v \in P(s, p) \setminus \{p\}$.*
2. *$y'_v = y_v + \epsilon$ for all $v \in P(t, p) \setminus \{p\}$.*
3. *$y'_v = y_v$ for all other $v \in V(\mathcal{T})$.*

Furthermore, we say that y, y' are strictly ϵ-neighboring if ϵ is sufficiently small to satisfy

1. *$\epsilon \le \delta(y_v)$ for all $v \in P(s, p) \setminus \{p\}$ with $\delta(y_v) > 0$.*
2. *$\epsilon \le 1 - \delta(y_v)$ for all $v \in P(t, p) \setminus \{p\}$ with $\delta(y_v) > 0$.*
3. *$\epsilon < 1$.*

Basically, if y and y' are ϵ-neighboring then y' is obtained by pushing ϵ-mass on y from s to t along the unique path that connects these two leaves. Furthermore, if ϵ is sufficiently small so that for any $v \in V(\mathcal{T})$ either $\lfloor y_v \rfloor = \lfloor y'_v \rfloor$ or $|y_v - y'_v| \le 1$ and at least one of the two is integral, then we say that the two fractional facility placements are *strictly* ϵ-neighboring. As we will shortly argue, Lemma 7 holds in the special case where y, y' are strictly ϵ-neighboring.

Claim 17. *If $y, y' \in \mathcal{FP}(\mathcal{T})$ are strictly ϵ-neighboring for some $\epsilon \ge 0$, then*

$$\gamma \cdot \mathbb{E}_\alpha \left[M_\mathcal{T}(F, F')\right] \le 4 \cdot ||y - y'||_\mathcal{T}.$$

Before proving Claim 17, let us first show why it suffices to argue about the general case and prove Lemma 7. Let $y, y' \in \mathcal{FP}(\mathcal{T})$ be any two fractional placements. Recall that $||y - y'||_\mathcal{T}$ captures precisely the minimum transportation cost from y to y' on $\mathcal{T}$. If we break down this transportation plan into small movements of masses between leaves, then we can view it as a sequence of transitions between strictly ϵ-neighboring placements. This is formalized in the following claim:

Claim 18. *For any $y, y' \in \mathcal{FP}(\mathcal{T})$, there exists a finite sequence $y_0, y_1, \ldots, y_m \in \mathcal{FP}(\mathcal{T})$ of fractional facility placements with $y = y_0$ and $y' = y_m$ such that*

1. *y_j, y_{j+1} are strictly ϵ-neighboring for some $\epsilon \geq 0$ for $j = 0, 1, \ldots, m-1$.*
2. *$||y - y'||_{\mathcal{T}} = \sum_{j=1}^{m} ||y_j - y_{j-1}||_{\mathcal{T}}$.*

We will now prove Lemma 7. Let $F_j = \text{Cut\&Round}(\mathcal{T}, y_j, \alpha)$ be the corresponding output of $\text{Cut\&Round}$ on y_j using the same (uniformly random) thresholds α_v. Then,

$$\begin{aligned}
\gamma \cdot \mathbb{E}_\alpha \left[M_{\mathcal{T}}(F, F')\right] &\leq \gamma \cdot \mathbb{E}_\alpha \left[\sum_{j=0}^{m-1} M_{\mathcal{T}}(F_j, F_{j+1})\right] \\
&= \gamma \cdot \sum_{j=0}^{m-1} \mathbb{E}_\alpha \left[M_{\mathcal{T}}(F_j, F_{j+1})\right] \\
&\leq 4 \cdot \sum_{j=0}^{m-1} ||y_j - y_{j+1}||_{\mathcal{T}} \\
&= 4 \cdot ||y - y'||_{\mathcal{T}}
\end{aligned}$$

In the above calculation, the first inequality holds from the fact that the minimum transportation cost satisfies the triangular inequality. The first equality holds from linearity of expectation. The second inequality holds from Claim 17 and the second equality holds from Claim 18.

Thus, we have shown that proving Lemma 7 for the special case of strictly ϵ-neighboring fractional facility placements y, y' suffices to prove Lemma 7 for the general case of any $y, y' \in \mathcal{FP}(\mathcal{T})$ and conclude this section. The rest of this section is dedicated to proving Claim 17, which is the main technical challenge towards proving Lemma 7.

Proof of Claim 17. Fix any pair of strictly ϵ-neighboring fractional facility placements $y, y' \in \mathcal{FP}(\mathcal{T})$ and let the corresponding outputs of $\text{Cut\&Round}$ be $F = \text{Cut\&Round}(\mathcal{T}, y, \alpha)$ and $F' = \text{Cut\&Round}(\mathcal{T}, y', \alpha)$. In section E.1 we have already shown that $F, F' \subseteq L(\mathcal{T})$ are valid facility placements since $|F| = |F'| = k$. Let $Y, Y' \in \mathcal{FP}(\mathcal{T})$ be used to denote the corresponding integral placements, i.e.

$Y_v := |L(\mathcal{T}) \cap F| =$ number of facilities in F placed on the leaves of the sub-tree rooted at v

and

$Y'_v := |L(\mathcal{T}) \cap F'| =$ number of facilities in F' placed on the leaves of the sub-tree rooted at v

Recall that Y and Y' are precisely the values of the Y-variables that algorithm $\text{Cut\&Round}$ sets. As shown in Claim 2 of Section E.4, we know that $\gamma \cdot M_{\mathcal{T}}(F, F') = ||Y - Y'||_{\mathcal{T}}$. Thus, in order to prove Claim 17, we need to show that

$$\mathbb{E}_\alpha \left[||Y - Y'||_{\mathcal{T}}\right] \leq 4 \cdot ||y - y'||_{\mathcal{T}}$$

Since y, y' are strictly ϵ-neighboring fractional facility placements, we know that there exist two leaves $s, t \in L(\mathcal{T})$ with lowest common ancestor $p \in V(\mathcal{T})$ such that $|y_v - y'_v|$ is ϵ among vertices on the (unique) path from s to t (excluding vertex p) and is 0 otherwise. Let $L = \text{lev}(p)$. Then, by definition of $|| \cdot ||_{\mathcal{T}}$ we have

$$||y - y'||_{\mathcal{T}} = \sum_{v \in V(\mathcal{T})} 2^{\text{lev}(v)} \cdot |y_v - y'_v| = 2\epsilon \cdot \sum_{l=0}^{L-1} 2^l = 2\epsilon \cdot (2^L - 1) \tag{5}$$

Furthermore, recall that from Lemma 5, $\text{Cut\&Round}$ rounds y_v to either $Y_v = \lfloor y_v \rfloor + 1$ with probability $\delta(y_v)$ or to $\lfloor y_v \rfloor$ with probability $1 - \delta(y_v)$. Since ϵ is sufficiently small so that either $\lfloor y_v \rfloor = \lfloor y'_v \rfloor$ or $|y_v - y'_v| \leq 1$ and at lowest one of the two is integral (and it is thus always rounded to itself), we get that $|Y_v - Y'_v| \leq 1$ for all $v \in V(\mathcal{T})$. This implies that

$$\begin{aligned}
\mathbb{E}_\alpha\left[||Y - Y'||_{\mathcal{T}}\right] &= \mathbb{E}_\alpha\left[\sum_{v \in V(\mathcal{T})} 2^{\mathrm{lev}(v)} \cdot |Y_v - Y'_v|\right] \\
&= \sum_{v \in V(\mathcal{T})} 2^{\mathrm{lev}(v)} \cdot \mathbb{E}_\alpha\left[|Y_v - Y'_v|\right] \\
&= \sum_{v \in V(\mathcal{T})} 2^{\mathrm{lev}(v)} \cdot \mathbb{P}_\alpha\left[|Y_v - Y'_v| = 1\right]
\end{aligned}$$

Let $l \in [0, h(\mathcal{T})]$ be any level on the HST $\mathcal{T}$ and let C_l be used to denote the expected number of vertices at level l that are rounded to different values, i.e.

$$C_l := \mathbb{E}_\alpha\left[|\{v \in V(\mathcal{T}) : \mathrm{lev}(v) = l \text{ and } Y_v \neq Y'_v\}|\right]$$

Then, the above imply that

$$\mathbb{E}_\alpha\left[||Y - Y'||_{\mathcal{T}}\right] = \sum_{l=0}^{h(\mathcal{T})} 2^l \cdot C_l \tag{6}$$

It remains to compute C_l for all $l \in [0, h(\mathcal{T})]$. This is done in Claim 19, where we prove that $C_l = 0$ for $l \geq L$ (the level of s and t's lowest common ancestor) and $C_l \leq 4\epsilon \cdot (L - l)$ otherwise. Combining this claim with equations (5) and (6) immediately implies that

$$\mathbb{E}_\alpha\left[||Y - Y'||_{\mathcal{T}}\right] \leq 4 \cdot ||y - y'||_{\mathcal{T}}$$

which completes the proof of Claim 17.

Claim 19. *For any $l \geq L$, $C_l = 0$. For any $l < L$, $C_l \leq 4\epsilon \cdot (L - l)$.*

Proof. Recall that for fixed thresholds α_v, the output of Cut&Round is deterministic. Since L is the level of vertex p (the lowest common ancestor of leaves s, t) and by definition of strictly ϵ-neighboring placements y, y' we know $y_v = y'_v$ for any vertex v such that $\mathrm{lev}(v) \geq L$, we immediately get that $C_l = 0$ for any $l \geq L$.

We will now proceed to analyze C_l for any $l < L$. We partition the set of vertices $v \in V(\mathcal{T})$ with $\mathrm{lev}(v) = l$ into three sets:

- A vertex v is called *active* if it lies on the (unique) path between leaves s and t.
- A vertex v is called *inactive* if it is not a descendant of p (the lowest common ancestor of leaves s and t).
- A vertex v is called *affected* if it is not active and is a descendant of p.

Obviously, each vertex v with $\mathrm{lev}(v) = l$ must lie in exactly one of these sets.

Inactive Vertices. We will prove that for every inactive vertex v, $\mathbb{P}_\alpha[Y_v \neq Y'_v] = 0$. Since the same set of thresholds α is used to round both y and y', the output of Cut&Round is deterministic. Furthermore, if a vertex v is inactive, then we know that $y_v = y'_v$ and also $y_u = y'_u$ for any ancestor vertex of u of v (by Definition 10 of neighboring facility placements). Thus, this immediately implies that $Y_v = Y'_v$ with probability 1 and thus we do not need to account for inactive vertices when computing C_l.

Active Vertices. We will prove that for every active vertex v, $\mathbb{P}_\alpha[Y_v \neq Y'_v] = \epsilon$. Recall that any active vertex is either an ancestor of leaf s or leaf t. We will only prove the claim in the case when v is an ancestor of t; the other case is completely analogous. A formal proof by induction is given in Claim 20, presented at the end of this section. As a direct corollary, since there are only two active vertices per level, the expected number of active vertices in level l that are rounded two different values is precisely 2ϵ.

Affected Vertices. Finally, we will now analyze the affected vertices. By definition, we know that each affected vertex v will have a unique active ancestor (also counting p). We partition the set of affected vertices on level l into $2(L-l-1)+1$ groups, based on their corresponding active ancestor. The main argument we need to establish is that by definition of $\mathrm{Round\&Cut}$, at most one vertex in each of these groups can be rounded to a different value.

To see this, observe that $\mathrm{Round\&Cut}$ is monotone, in the sense that if $y'_v \geq y_v$ and also $y'_u \geq y_u$ for all ancestors u of v, then (assuming the same set of thresholds is used), we know that $Y'_v \geq Y_v$. Using this fact on the vertices of a group, since all of them can either only increase or decrease, in order to maintain balance at most one of them can change, otherwise we would get a change of 2 or more on the parent node which cannot happen.

Furthermore, for a specific group, if both the common active ancestor and its child u with $y_u \neq y'_u$ end up rounded to the same value, we get (from the fact that the same thresholds are used) that all the vertices in the group will be rounded to the same value. Thus, in order for a (unique) vertex in any group to change, at least one of two active vertices must change, which happens with probability at most 2ϵ. Since there are $2(L-l-1)+1$ groups, we get as a corollary that the expected number of affected vertices at level l that get rounded to a different value is at most $2\epsilon \cdot (2L-2l-1)$.

Combining everything, we get that $C_l \leq 0 + 2\epsilon + 2\epsilon \cdot (2L-2l-1) = 4\epsilon \cdot (L-l)$. ☐

Claim 20. *Let v be any active vertex that is an ancestor of t. Then, $\mathbb{P}_\alpha[Y_v \neq Y'_v] = \epsilon$.*

Proof. In fact, we will in fact prove the following stronger claim,

- $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor \text{ and } Y'_v = \lfloor y_v \rfloor] = 1 - \delta(y_v) - \epsilon$.
- $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor \text{ and } Y'_v = \lfloor y_v \rfloor + 1] = \epsilon$.
- $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor + 1 \text{ and } Y'_v = \lfloor y_v \rfloor] = 0$
- $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor + 1 \text{ and } Y'_v = \lfloor y_v \rfloor + 1] = \delta(y_v)$

which clearly implies Claim 20.

Once again, we will prove the claim via induction, starting from the highest active ancestor of t at level $l = L-1$ and moving towards the leaf t at level $l = 0$. We begin by mentioning that for vertex p (s and t's lowest common ancestor at level L) we know for sure that $Y_p = Y'_p$ since $y_p = y'_p$ and $y_u = y'_u$ for any u such that $\mathrm{lev}(u) \geq L$; thus, since the same set of thresholds α is used, the execution of $\mathrm{Cut\&Round}$ will be identical up to this point.

We assume that the first child of any vertex v visited by Alloc is always the active child; this can be done without loss of generality as the order that Alloc visits the vertices hasn't played any part on our analysis yet.

Base of the induction. For the base of the induction, let v be the (unique) child of p that is an ancestor of t; i.e. let v be the highest active ancestor of t. We have already mentioned that $Y_p = Y'_p$ with probability 1. Thus, it can either be the case that $Y_p = Y'_p = \lfloor y_p \rfloor$ of $Y_p = Y'_p = \lfloor y_p \rfloor + 1$. From Lemma 5 we know that the first happens with probability $1 - \delta(y_p)$ and the latter with probability $\delta(y_p)$. We distinguish between the following cases:

- Let $\delta(y_v) < \delta(y_p)$. Then, if $Y_p = Y'_p = \lfloor y_p \rfloor$ we know from the description of Alloc that $Y_v = Y'_v = \lfloor y_v \rfloor$ with probability 1. On the other hand, if $Y_p = Y'_p = \lfloor y_p \rfloor + 1$, we know that $Y_v = \lfloor y_v \rfloor + 1$ if $\alpha_v \leq \delta(y_v)/\delta(y_p)$ and likewise $Y'_v = \lfloor y_v \rfloor + 1$ if $\alpha_v \leq (\delta(y_v)+\epsilon)/(\delta(y_p))$. Thus, by conditioning on the values of Y_p and Y'_p, we get
  1. $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor \text{ and } Y'_v = \lfloor y_v \rfloor] = (1-\delta(y_p)) \cdot 1 + \delta(y_p) \cdot (1 - \frac{\delta(y_v)+\epsilon}{\delta(y_p)}) = 1 - \delta(y_v) - \epsilon$.
  2. $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor \text{ and } Y'_v = \lfloor y_v \rfloor + 1] = (1-\delta(y_p)) \cdot 0 + \delta(y_p) \cdot (\frac{\delta(y_v)+\epsilon}{\delta(y_p)} - \frac{\delta(y_v)}{\delta(y_p)}) = \epsilon$.
  3. $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor + 1 \text{ and } Y'_v = \lfloor y_v \rfloor] = (1-\delta(y_p)) \cdot 0 + \delta(y_p) \cdot 0 = 0$.

4. $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor + 1 \text{ and } Y'_v = \lfloor y_v \rfloor + 1] = (1 - \delta(y_p)) \cdot 0 + \delta(y_p) \cdot \frac{\delta(y_v)}{\delta(y_p)} = \delta(y_v)$.

- Let $\delta(y_v) \geq \delta(y_p)$. Then, if $Y_p = Y'_p = \lfloor y_p \rfloor + 1$ we know from the description of Alloc that $Y_v = Y'_v = \lfloor y_v \rfloor + 1$ with probability 1. On the other hand, if $Y_p = Y'_p = \lfloor y_p \rfloor$, we know that $Y_v = \lfloor y_v \rfloor + 1$ if $\alpha_v \leq (\delta(y_v) - \delta(y_p))/(1 - \delta(y_p))$ and likewise $Y'_v = \lfloor y_v \rfloor + 1$ if $\alpha_v \leq (\delta(y_v) + \epsilon - \delta(y_p))/(1 - \delta(y_p))$. Thus, by conditioning on the values of Y_p and Y'_p, we get

  1. $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor \text{ and } Y'_v = \lfloor y_v \rfloor] = (1 - \delta(y_p)) \cdot (1 - \frac{\delta(y_v)+\epsilon-\delta(y_p)}{1-\delta(y_p)}) + \delta(y_p) \cdot 0 = 1 - \delta(y_v) - \epsilon$.
  2. $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor \text{ and } Y'_v = \lfloor y_v \rfloor] = (1 - \delta(y_p)) \cdot (\frac{\delta(y_v)+\epsilon-\delta(y_p)}{1-\delta(y_p)} - \frac{\delta(y_v)-\delta(y_p)}{1-\delta(y_p)}) + \delta(y_p) \cdot 0 = \epsilon$.
  3. $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor + 1 \text{ and } Y'_v = \lfloor y_v \rfloor] = (1 - \delta(y_p)) \cdot 0 + \delta(y_p) \cdot 0 = 1 - \delta(y_v) = 0$.
  4. $\mathbb{P}_\alpha[Y_v = \lfloor y_v \rfloor + 1 \text{ and } Y'_v = \lfloor y_v \rfloor] = (1 - \delta(y_p)) \cdot \frac{\delta(y_v)-\delta(y_p)}{1-\delta(y_p)} + \delta(y_p) \cdot 1\delta(y_v)$.

So in both cases, the base of the induction holds.

Inductive Step. Using the exact same approach, we can prove the claim for any active ancestor u of t, assuming that the claim holds for u's father $v = p(u)$. The only difference, is that now we can't claim that $Y_v = Y'_v$ with probability 1. Instead, there are three different cases that we need to consider; namely

1. $Y_v = Y'_v = \lfloor y_v \rfloor$ with probability $1 - \epsilon - \delta(y_v)$.
2. $Y_v = Y'_v = \lfloor y_v \rfloor + 1$ with probability $\delta(y_v)$.
3. $Y_v = \lfloor y_v \rfloor$ and $Y'_v = \lfloor y_v \rfloor + 1$ with probability ϵ.

where the probabilities hold from the inductive hypothesis on the parent vertex v. Next, we will need to once again consider the case of whether $\delta(y_u) < \delta(y_v)$ or not (notice that the same relation will hold for y'_u and y'_v) and use the description of Alloc to get the assignment probabilities. Since this is a simple matter of arithmetic, the details are omitted. $\square$

F Experimental Evaluation

In this section we experimentally evaluate the performance of Algorithm 2 with respect to the best fixed facility placement and compare it with the respective performance of the algorithm proposed by [30]. In all the following experiments the step-size of Algorithm 3 (subroutine of Algorithm 2) is set to $\eta := \max(\gamma, 1)\sqrt{nT}$. All the code was written in Python and is included in the supplementary material.

Periodically Moving Clients. We first present a simple setting to indicate the inefficiency of the online learning algorithm of [30] in handling moving costs. In this experiment the underlying graph is the 0.01-discretization of $[0, 1] \times [0, 1]$. At each round $t \geq 1$, we periodically select one of four balls of radius $R = 0.2$ depicted in Figure 1 and then a client arrives uniformly at random on the selected ball. In Figure 1 and Table 1 we present the overall cost of Algorithm 2 and the algorithm of [30] for different values of facility-weight γ, $k = 3$ facilities and $T = 4000$ time-steps. In all cases, the facilities of Algorithm 2 eventually converge to three of the four ball-centers, which is the optimal fixed facility placement. As the experiment reveals, the algorithm of [30] admits significantly larger cost as the facility-weight increases while Algorithm 2 is robust to the increase.

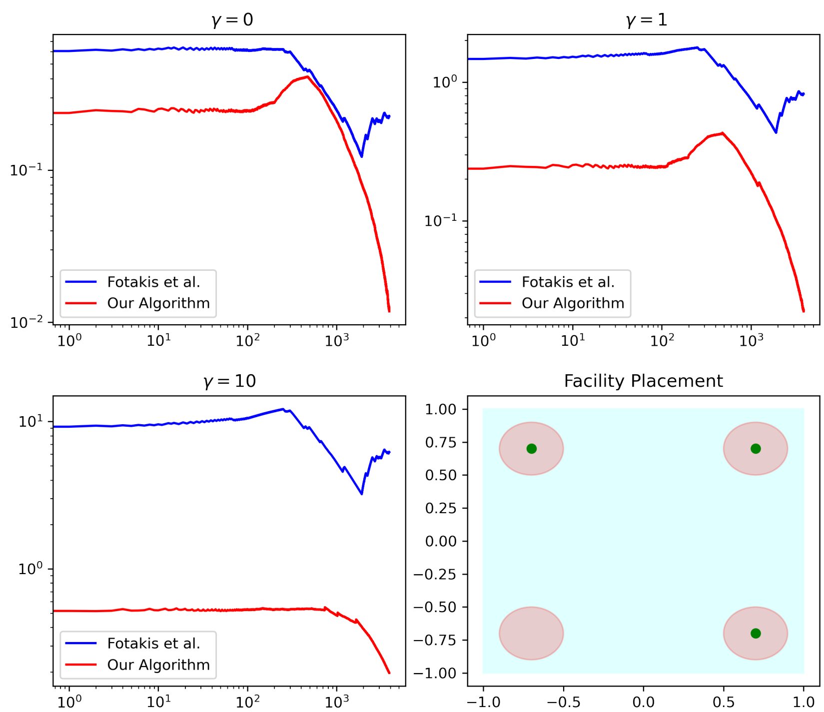

Figure 1: We plot the evolution of the approximation ratio for Algorithm 2 (red curve) and the algorithm from [30] (blue curve) compared to the hindsight optimal facility placement for facility weights $\gamma = 0, \gamma = 1$ and $\gamma = 10$. Both scales are logarithmic. The bottom-right plot depicts the facilities eventually placed by our Algorithm 2 which coincides with the optimal configuration

Table 1: Ratio of the overall cost of both algorithms with respect to to the hindsight optimal (20 runs).

MovingClients	$\gamma = 0$	$\gamma = 1$	$\gamma = 10$
[30]	1.297 ± 0.045	1.943 ± 0.466	3.388 ± 1.335
Algorithm 2	1.083 ± 0.001	1.091 ± 0.001	1.343 ± 0.014

Real-World Datasets. We evaluate the performance of Algorithm 2 on the MNIST and $\mathrm{CIFAR10}$ datasets. We randomly sample $N = 10000$ images and construct a graph where each image corresponds to a vertex with the edge weights given by the Euclidean distance of the respective images. At each round t, an image is sampled uniformly at random and a client arrives in the corresponding vertex. We then evaluate Algorithm 2 in the latter setting for $T = 3000$ rounds and $k = 10$ facilities. In Table 2 we present the ratio of the overall cost of Algorithm 2 over the ratio cost of the fractional hindsight optimal[5]. As our experiments indicate, the sub-optimality of Algorithm 2 is way smaller than the theoretical $\mathcal{O}(\log n)$ upper bound on the regret.

Table 2: The ratio of the cost of Algorithm 2 with respect to the cost of the fractional hindsight optimal facility placement (20 runs).

Algorithm 2	$\gamma = 0$	$\gamma = 1$	$\gamma = 10$
MNIST	1.118 ± 0.01	1.403 ± 0.04	1.5631 ± 0.03
CIFAR10	1.113 ± 0.01	1.189 ± 0.04	1.59 ± 0.31

[5]The cost of the fractional hindsight optimal can be efficiently computed [30] and lower bounds the cost of the optimal facility placement. As a result, the presented ratios in Table 2 are upper bounds on the actual ratio of Algorithm 2 and the optimal facility-placement.