Pure Entropic Regularization for Metrical Task Systems

Christian Coester CHRISTIAN.COESTER@CS.OX.AC.UK
University of Oxford

James R. Lee JRL@CS.WASHINGTON.EDU
University of Washington

Editors: Alina Beygelzimer and Daniel Hsu

Abstract

We show that on every n-point HST metric, there is a randomized online algorithm for metrical task systems (MTS) that is 1-competitive for service costs and $O(\log n)$-competitive for movement costs. In general, these refined guarantees are optimal up to the implicit constant. While an $O(\log n)$-competitive algorithm for MTS on HST metrics was developed by Bubeck et al. (2018a), that approach could only establish an $O((\log n)^2)$-competitive ratio when the service costs are required to be $O(1)$-competitive. Our algorithm is an instantiation of online mirror descent with the regularizer derived from a multiscale conditional entropy.

In fact, our algorithm satisfies a set of even more refined guarantees; we are able to exploit this property to combine it with known random embedding theorems and obtain, for *any* n-point metric space, a randomized algorithm that is 1-competitive for service costs and $O((\log n)^2)$-competitive for movement costs.

Keywords: Online algorithms, competitive analysis, mirror descent, metrical task systems, decision making under uncertainty

1. Introduction

Let (X, d_X) be a finite metric space with $|X| = n > 1$. The Metrical Task Systems (MTS) problem, introduced by Borodin et al. (1992) is described as follows. The input is a sequence $\langle c_t : X \to \mathbb{R}_+ : t \geq 1\rangle$ of nonnegative cost functions on the state space X. At every time t, an online algorithm maintains a state $\rho_t \in X$.

The corresponding cost is the sum of a *service cost* $c_t(\rho_t)$ and a *movement cost* $d_X(\rho_{t-1}, \rho_t)$. Formally, an *online algorithm* is a sequence of mappings $\boldsymbol{\rho} = \langle \rho_1, \rho_2, \ldots, \rangle$ where, for every $t \geq 1$, $\rho_t : (\mathbb{R}_+^X)^t \to X$ maps a sequence of cost functions $\langle c_1, \ldots, c_t\rangle$ to a state. The initial state $\rho_0 \in X$ is fixed. The *total cost of the algorithm* $\boldsymbol{\rho}$ *in servicing* $\boldsymbol{c} = \langle c_t : t \geq 1\rangle$ is defined as:

$$\mathrm{cost}_{\boldsymbol{\rho}}(\boldsymbol{c}) := \sum_{t\geq 1} \left[c_t\big(\rho_t(c_1, \ldots, c_t)\big) + d_X\big(\rho_{t-1}(c_1, \ldots, c_{t-1}), \rho_t(c_1, \ldots, c_t)\big)\right].$$

The cost of the *offline optimum*, denoted $\mathrm{cost}^*(\boldsymbol{c})$, is the infimum of $\sum_{t\geq 1}[c_t(\rho_t) + d_X(\rho_{t-1}, \rho_t)]$ over *any* sequence $\langle \rho_t : t \geq 1\rangle$ of states. A *randomized online algorithm* $\boldsymbol{\rho}$ is said to be α-*competitive* if for every $\rho_0 \in X$, there is a constant $\beta > 0$ such that for all cost sequences $\boldsymbol{c}$:

$$\mathbb{E}\left[\mathrm{cost}_{\boldsymbol{\rho}}(\boldsymbol{c})\right] \leq \alpha \cdot \mathrm{cost}^*(\boldsymbol{c}) + \beta\,.$$

For the n-point uniform metric, a simple coupon-collector argument shows that the competitive ratio is $\Omega(\log n)$, and this is tight (Borodin et al. (1992)). A long-standing conjecture is that this

$\Theta(\log n)$ competitive ratio holds for an arbitrary n-point metric space. The lower bound has almost been established in Bartal et al. (2006, 2005); for any n-point metric space, the competitive ratio is $\Omega(\log n/\log\log n)$. Following a long sequence of works (see, e.g., Seiden (1999); Blum et al. (2000); Bartal et al. (1997); Bartal (1996); Fiat and Mendel (2003); Fakcharoenphol et al. (2004)), an upper bound of $O((\log n)^2)$ was shown in Bubeck et al. (2018a).

Relation to adversarial multi-arm bandits MTS is naturally related to the adversarial setting of the classical multi-arm bandits model in sequential decision making, and provides a very general framework for "bandits with switching costs." Unlike in the setting of regret minimization, where one competes against the best static strategy in hindsight (see, e.g., Bubeck and Cesa-Bianchi (2012)), competitive analysis compares the performance of an online algorithm to the best *dynamical* offline algorithm.

Thus this model emphasizes the importance of an adaptivity in the face of changing environments. For MTS, the online algorithm has *full information*: access to the complete cost function c_t is available when deciding on a point $\rho_t(c_1, \dots, c_t) \in X$ at which to play. And yet one of the fascinating relationships between MTS and adversarial bandits is the parallel between adaptivity—being willing to "try out" new strategies—and the classical exploration/exploitation tradeoff that occurs in models where one only has access to partial information about the loss functions.

HST metrics The methods of Bansal et al. (2012) show that the competitive ratio for MTS is $O(\log n)$ on weighted star metrics. Recently, Bubeck et al. (2018a) generalized this result by designing an algorithm with competitive ratio $O(\mathfrak{D}_T \log n)$ on any weighted n-point tree metric with combinatorial depth $\mathfrak{D}_T$. We now discuss a special class of metrics.

Let $T = (V, E)$ be a finite tree with root $\mathbb{r}$ and vertex weights $\{w_u > 0 : u \in V\}$, let $\mathcal{L} \subseteq V$ denote the leaves of T, and suppose that the vertex weights on T are non-increasing along root-leaf paths. Consider the metric space $(\mathcal{L}, d_T)$, where $d_T(\ell, \ell')$ is the weighted length of the path connecting ℓ and ℓ' when the edge from a node u to its parent is w_u. We will use $\mathfrak{D}_T$ for the combinatorial (i.e., unweighted) depth of T.

$(\mathcal{L}, d_T)$ is called an *HST metric* (or, equivalently for finite metric spaces, an *ultrametric*). If, for some $\tau > 1$, the weights on T satisfy the stronger inequality $w_v \le w_u/\tau$ whenever v is a child of u, the space $(\mathcal{L}, d_T)$ is said to be a τ-*HST metric.* Such metric spaces play a special role in MTS since every n-point metric space can be probabilistically approximated by a distribution over such spaces (Bartal (1996); Fakcharoenphol et al. (2004)). Indeed, the $O((\log n)^2)$-competitive ratio for general metric spaces established in Bubeck et al. (2018a) is a consequence of their $O(\log n)$-competitive algorithm for HSTs.

1.1. Refined guarantees

The authors of Bansal et al. (2010) observe that there is a more refined way to analyze competive algorithms for MTS. For a randomized online algorithm $\boldsymbol{\rho}$ and a cost sequence $\boldsymbol{c}$, we denote, respectively, $\mathsf{S}_\rho(\boldsymbol{c})$ and $\mathsf{M}_\rho(\boldsymbol{c})$ for the (expected) service cost and movement cost, that is

$$\mathsf{S}_\rho(\boldsymbol{c}) := \mathbb{E}\sum_{t\ge 1} c_t(\rho_t) \quad \text{and} \quad \mathsf{M}_\rho(\boldsymbol{c}) := \mathbb{E}\sum_{t\ge 1} d_X(\rho_{t-1}, \rho_t)\,.$$

If there are numbers $\alpha, \alpha', \beta, \beta' > 0$ such that for every cost $\boldsymbol{c}$, it holds that

$$\mathsf{S}_{\boldsymbol{\rho}}(\boldsymbol{c}) \leq \alpha \cdot \mathrm{cost}^*(\boldsymbol{c}) + \beta$$
$$\mathsf{M}_{\boldsymbol{\rho}}(\boldsymbol{c}) \leq \alpha' \cdot \mathrm{cost}^*(\boldsymbol{c}) + \beta',$$

one says that $\boldsymbol{\rho}$ is *α-competitive for service costs* and *α'-competitive for movement costs.*

In Bansal et al. (2010), it is shown that on every n-point HST metric, and for every $\epsilon > 0$, there is an online algorithm that is simultaneously $(1+\epsilon)$-competitive for service costs and $O((\log(n/\epsilon))^2)$-competitive for movement costs. Bubeck et al. (2018a) improve this slightly to show that actually there is an online algorithm that is simultaneously 1-competitive for service costs and $O((\log n)^2)$-competitive for movement costs. We obtain the optimal refined guarantees.

Theorem 1 *On any n-point HST metric X, there is a randomized online algorithm that is 1-competitive for service costs and $O(\log n)$-competitive for movement costs.*

Remark 2 (Optimality of the refined guarantees) *It is not difficult to see that by making the costs sufficiently large, any finitely competitive algorithm for MTS on an n-point uniform metric cannot be better than $\Omega(\log n)$-competitive for movement costs, regardless of its competitive ratio for service costs. Moreover, it cannot be better than 1-competitive for service costs, regardless of its competitive ratio for movement costs.*

Finely competitive guarantees Suppose that for some numbers $\alpha_0, \alpha_1, \gamma, \beta, \beta' > 0$, a randomized online algorithm $\boldsymbol{\rho}$ satisfies, for every cost $\boldsymbol{c}$ and *every* offline algorithm $\boldsymbol{\rho}^*$:

$$\mathsf{S}_{\boldsymbol{\rho}}(\boldsymbol{c}) \leq \alpha_0 \mathsf{S}_{\boldsymbol{\rho}^*}(\boldsymbol{c}) + \alpha_1 \mathsf{M}_{\boldsymbol{\rho}^*}(\boldsymbol{c}) + \beta \tag{1}$$
$$\mathsf{M}_{\boldsymbol{\rho}}(\boldsymbol{c}) \leq \gamma \mathsf{S}_{\boldsymbol{\rho}}(\boldsymbol{c}) + \beta' . \tag{2}$$

In this case, we say that $\boldsymbol{\rho}$ *is $(\alpha_0, \alpha_1, \gamma)$-finely competitive.* We establish the following.

Theorem 3 *On any n-point HST metric X, for every $\kappa \geq 1$, there is an online randomized algorithm $\boldsymbol{\rho}$ that is $(1, 1/\kappa, O(\kappa \log n))$-finely competitive. In fact, one can take $\beta = 0$ and $\beta' \leq O(\kappa \mathrm{diam}(X))$.*

Combined with the random embedding from Fakcharoenphol et al. (2004), this yields the following consequence for general n-point metric spaces.

Corollary 4 *On any n-point metric space, there is an online randomized algorithm that is 1-competitive for service costs and $O((\log n)^2)$-competitive for movement costs.*

Proof Consider an n-point metric space (X, d_X). It is known (Fakcharoenphol et al. (2004)) that there exists a random HST metric (T, d_T) so that $\mathcal{L}(T) = X$ and for all $x, y \in X$:

1. $\Pr[d_T(x,y) \geq d_X(x,y)] = 1$,
2. $\mathbb{E}[d_T(x,y)] \leq D \cdot d_X(x,y)$,

and $D \leq O(\log n)$.

Let $\boldsymbol{\rho}_T$ be the randomized algorithm for (T, d_T) guaranteed by Theorem 3 with $\kappa = D$. Let $\boldsymbol{\rho}$ denote the algorithm that results from sampling (T, d_T) and then using $\boldsymbol{\rho}_T$. We use M^T to denote movement cost measured in d_T and M^X for movement cost measured in d_X.

Then for any cost $\boldsymbol{c}$ and any offline algorithm $\boldsymbol{\rho}^*$, we have

$$\begin{aligned}\mathsf{S}_{\boldsymbol{\rho}}(\boldsymbol{c}) = \mathbb{E}[\mathsf{S}_{\boldsymbol{\rho}_T}(\boldsymbol{c})] &\leq \mathsf{S}_{\boldsymbol{\rho}^*}(\boldsymbol{c}) + \kappa^{-1}\,\mathbb{E}[\mathsf{M}^T_{\boldsymbol{\rho}^*}(\boldsymbol{c})] + O(1) \\ &\leq \mathsf{S}_{\boldsymbol{\rho}^*}(\boldsymbol{c}) + \kappa^{-1} D \mathsf{M}^X_{\boldsymbol{\rho}^*}(\boldsymbol{c}) + O(1) \\ &= \mathsf{S}_{\boldsymbol{\rho}^*}(\boldsymbol{c}) + \mathsf{M}^X_{\boldsymbol{\rho}^*}(\boldsymbol{c}) + O(1)\,,\end{aligned}$$

and

$$\mathsf{M}^X_{\boldsymbol{\rho}}(\boldsymbol{c}) = \mathbb{E}[\mathsf{M}^X_{\boldsymbol{\rho}_T}(\boldsymbol{c})] \leq \mathbb{E}[\mathsf{M}^T_{\boldsymbol{\rho}_T}(\boldsymbol{c})] \leq O(\kappa \log n)\,\mathbb{E}[\mathsf{S}_{\boldsymbol{\rho}_T}(\boldsymbol{c})] + O(1),$$

completing the proof. ∎

1.2. The fractional model on trees

We will work in the following deterministic fractional setting, which is equivalent to the randomized integral setting described earlier (see (Bubeck et al., 2018a, §2)). The state of a fractional algorithm is given by a point in the polytope

$$\mathsf{K}_T := \left\{ x \in \mathbb{R}^V_+ : x_{\mathbb{r}} = 1,\; x_u = \sum_{v \in \chi(u)} x_v \quad \forall u \in V \setminus \mathcal{L} \right\}, \tag{3}$$

where we use $\chi(u)$ for the set of children of u in T. For $u \neq \mathbb{r}$, we will also write $\mathsf{p}(u)$ for the parent of u in T.

A state $x \in \mathsf{K}_T$ corresponds to the situation that the state of a randomized integral algorithm is a leaf descendant of u with probability x_u. Note that K_T is simply an affine encoding of the probability simplex on $\mathcal{L}$. In the fractional setting, changing from state x to x' incurs movement cost $\|x - x'\|_{\ell_1(w)}$, where

$$\|z\|_{\ell_1(w)} := \sum_{u \in V} w_u |z_u|$$

denotes the weighted ℓ_1-norm on $\mathbb{R}^V$.

1.3. Mirror descent, metric filtrations, and regularization

Following Bubeck et al. (2018a), our algorithm is based on the mirror descent framework as established in Bubeck et al. (2018b). This is a method for regularized online convex optimization, an approach that was previously explored for competitive analysis by Abernethy et al. (2010) and Buchbinder et al. (2014).

A central component of mirror descent is choosing the appropriate mirror map (which we will often refer to as the "regularizer"). This is a strictly convex function $\Phi : \mathsf{K}_T \to \mathbb{R}$ that endows

K_T with a geometric (Riemannian) structure, specifying how to perform constrained vector flow. In other words, it specifies how one can move in a preferred direction while remaining inside K_T.

The paper Bubeck et al. (2018a) employs the following regularizer:

$$\Phi_0(x) := \frac{1}{\eta} \sum_{u \in V \setminus \{\mathbb{r}\}} w_u \left(x_u + \delta_u\right) \log\left(x_u + \delta_u\right) \,, \tag{4}$$

with $\eta \asymp \log |\mathcal{L}|$ and $\delta_u = |\mathcal{L}_u|/|\mathcal{L}|$, where $\mathcal{L}_u$ is the set of leaves in the subtree rooted at u.

1.3.1. METRIC FILTRATIONS

It is straightforward that one can think of Φ_0 as a type of multiscale entropy (this is the *negative* of the associated Shannon entropy, since we use the analyst's convention that the entropy is convex). To understand this notion, let us forget momentarily the weights on T. Then the structure of T gives a natural filtration over probability measures on the leaves $\mathcal{L}$. Suppose that $\boldsymbol{X}$ is a random variable taking values in $\mathcal{L}$ and, for $u \in V$, denote $\boldsymbol{X}_u := \mathbb{1}_{\{\boldsymbol{X} \in \mathcal{L}_u\}}$. Then the chain rule for Shannon entropy yields

$$\sum_{\ell \in \mathcal{L}} \boldsymbol{X}_\ell \log \boldsymbol{X}_\ell = \sum_{u \in V \setminus \{\mathbb{r}\}} \boldsymbol{X}_u \log\left(\frac{\boldsymbol{X}_u}{\boldsymbol{X}_{\mathsf{p}(u)}}\right).$$

If we now imagine that uncertainty at higher scales is more costly than uncertainty at lower scales, then we might define an analogous *weighted* entropy by

$$\sum_{u \in V \setminus \{\mathbb{r}\}} w_u \boldsymbol{X}_u \log\left(\frac{\boldsymbol{X}_u}{\boldsymbol{X}_{\mathsf{p}(u)}}\right). \tag{5}$$

Such a notion is natural in the context of "metric learning" problems.

Ignoring the $\{\delta_u\}$ values for a moment, consider that (4) is not analogous to (5). Indeed, it corresponds to the quantity

$$\sum_{u \in V \setminus \{\mathbb{r}\}} w_u \boldsymbol{X}_u \log \boldsymbol{X}_u \,, \tag{6}$$

and now one can see a fundamental reason why the algorithm associated to (4) only achieves an $O(\mathfrak{D}_T \log n)$ competitive ratio, where $\mathfrak{D}_T$ is the combinatorial depth of T: The quantity (6) *over-measures* the metric uncertainty.

Suppose that $\boldsymbol{X}$ is a uniformly random leaf. Then $\sum_{\ell \in \mathcal{L}} \boldsymbol{X}_\ell \log \boldsymbol{X}_\ell = \log n$, where $n = |\mathcal{L}|$. But, in general, one could have $\sum_{u \in V} \boldsymbol{X}_u \log \boldsymbol{X}_u \geq \Omega(\mathfrak{D}_T \log n)$. Since the vertex weights are decreasing geometrically down root-leaf paths, the quantity (6) is actually within an $O(1)$ factor of (5), but given the manner in which the regularizer distorts the geometry, the overlap effect occurs as for the unweighted entropy. This fact was not lost on the authors of Bubeck et al. (2018a), but they bypass the problem by combining mirror descent on stars with a recursive composition method called "unfair gluing."

1.3.2. MULTISCALE CONDITIONAL ENTROPY

We employ a regularizer that is a more faithful analog of (5):

$$\Phi(x) := \sum_{u \in V \setminus \{\mathbb{r}\}} \frac{w_u}{\eta_u} \left(x_u + \delta_u x_{\mathsf{p}(u)}\right) \log\left(\frac{x_u}{x_{\mathsf{p}(u)}} + \delta_u\right), \tag{7}$$

where $\mathsf{p}(u)$ denotes the parent of u.

If one ignores the additional parameters $\{\eta_u \geq 1, \delta_u > 0\}$, this is precisely the negative weighted Shannon entropy written according to the chain rule. Here, we set

$$\theta_u := \frac{|\mathcal{L}_u|}{|\mathcal{L}_{\mathsf{p}(u)}|} \tag{8}$$

$$\eta_u := 1 + \log(1/\theta_u) \tag{9}$$

$$\delta_u := \theta_u/\eta_u \,. \tag{10}$$

The numbers $\{\theta_u\}$ are the conditional probabilites of the uniform distribution on leaves. The $\{\delta_u\}$ values are employed as "noise" added to the entropy calculation. Such noise is a fundamental aspect for competitive analysis, and distinguishes it from the application of mirror descent to regret minimization problems (see, e.g., Bubeck and Cesa-Bianchi (2012)).[1] The effect of these noise parameters appears ubiquitously in applications of the primal-dual method to competitive analysis (see Buchbinder and Naor (2007)), and manifests itself as an additive term in the update rules (see (11) below). Intuitively, it ensures that the conditional probability $\frac{x_u}{x_{\mathsf{p}(u)}}$ is updated fast enough even when it is close to 0.

Finally, the numbers $\{\eta_u : u \in V\}$ are commonly referred to as "learning rates" in the study of online learning. They represent the rate at which information is discounted in the resulting algorithm; for MTS, this corresponds to the relative importance of costs arriving now vs. costs that arrived in the past.

1.3.3. THE DYNAMICS

The resulting mirror descent algorithm $(x(t) \in \mathsf{K}_T : t \in [0, \infty))$ evolves naturally in continuous time:

$$\partial_t \left(\frac{x_u(t)}{x_{\mathsf{p}(u)}(t)}\right) = \frac{\eta_u}{w_u}\left(\frac{x_u(t)}{x_{\mathsf{p}(u)}(t)} + \delta_u\right)\left(\beta_{\mathsf{p}(u)}(t) - \sum_{\ell \in \mathcal{L}_u} \frac{x_\ell(t)}{x_u(t)} c_\ell(t)\right) \tag{11}$$

Here, $\beta_{\mathsf{p}(u)}(t)$ is a Lagrangian multiplier that ensures conservation of conditional probability:

$$\sum_{v \in \chi(\mathsf{p}(u))} \partial_t \left(\frac{x_v(t)}{x_{\mathsf{p}(u)}(t)}\right) = 0\,.$$

One can see that the evolution is being driven by the expected instantaneous cost incurred conditioned on the current state being in the subtree rooted at u.

One should interpret (11) only when $x(t)$ lies in the relative interior of K_T. Otherwise, the conditional probabilities are ill-defined. One way to rectify this is to prevent $x(t)$ from hitting the relative boundary of K_T at all. It is possible to adaptively modify the cost functions by a suitably small perturbation so as to guarantee this property and, at the same time, ensure that the total discrepancy between the modified and true service cost is a small additive constant.

Instead, we will follow a different approach, by extending the dynamics to an analogous system of conditional probabilities $\{q_u(t) : u \in V \setminus \{\mathbb{r}\}\}$:

$$\partial_t q_u(t) = \frac{\eta_u}{w_u}\left(q_u(t) + \delta_u\right)\left(\beta_{\mathsf{p}(u)}(t) - \hat{c}_u(t) + \alpha_u(t)\right), \tag{12}$$

1. One finds aspects of this "mixing with the uniform distribution" in the bandits setting as well, but used for variance reduction, a seemingly very different purpose.

where $q_u(t) = \frac{x_u(t)}{x_{\mathsf{p}(u)}(t)}$ whenever $x_{\mathsf{p}(u)}(t) > 0$, $\alpha_u(t)$ is a Lagrangian multiplier for the constraint $q_u(t) \geq 0$, and $\hat{c}_u(t)$ is the "derived" cost in the subtree rooted at u:

$$\hat{c}_u(t) := \sum_{\ell \in \mathcal{L}_u} q_{\ell|u}(t) c_\ell(t)$$

$$q_{\ell|u}(t) := \prod_{v \in \gamma_{u,\ell} \setminus \{u\}} q_v(t),$$

where $\gamma_{u,\ell}$ is the unique simple u-ℓ path in T.

Stated this way, the mirror descent algorithm can be envisioned as running a "weighted star" algorithm on the conditional probabilities at every internal node of T, with the derived costs at an internal node u given by the average cost of the current strategy for playing one unit of mass in the subtree rooted at u.

In the next section, we will implement and analyze a discretization of (12) using Bregman projections. Since our regularizer Φ and convex body K_T do not satisfy the assumptions underlying the existence and uniqueness theorem of Bubeck et al. (2018b), we need to construct a solution to (12) and, indeed, taking the discretization parameter in our algorithm to zero, one establishes an absolutely continuous solution. The major benefit of the formulations (11) and (12) is in motivating such an algorithm and prescribing the derived costs. We describe in the full version of our paper how these dynamics can be predicted from the definition (7).

2. Bregman projections

Consider a convex polytope $\mathsf{K}_0 \subseteq \mathbb{R}^n$, define $\mathsf{K} := \mathsf{K}_0 \cap \mathbb{R}^n_+$, and assume that K is compact. Suppose additionally that $\Phi : \mathcal{D} \to \mathbb{R}$ is differentiable and strictly convex in an open neighborhood $\mathcal{D} \supseteq \mathsf{K}$.

Let us write D_Φ for the corresponding Bregman divergence

$$\mathsf{D}_\Phi(y \,\|\, x) := \Phi(y) - \Phi(x) - \left\langle \nabla\Phi(x), y - x \right\rangle,$$

which is non-negative due to convexity of Φ. Then for $x, y, z \in \mathsf{K}$, we have:

$$\mathsf{D}_\Phi(z \,\|\, y) - \mathsf{D}_\Phi(z \,\|\, x) = -\Phi(y) + \Phi(x) - \langle \nabla\Phi(y), z - y\rangle + \langle \nabla\Phi(x), z - x\rangle. \tag{13}$$

For a vector $c \in \mathbb{R}^n$ and $x \in \mathsf{K}$, define the projection

$$\Pi^c_{\mathsf{K}}(x) := \operatorname{argmin}\left\{\mathsf{D}_\Phi(y \,\|\, x) + \langle c, y\rangle : y \in \mathsf{K}\right\}.$$

Since K is compact and Φ is strictly convex, there is a unique minimizer $y^* \in \mathsf{K}$.

For $x \in \mathsf{K}$, recall the definition of the normal cone at x:

$$\mathsf{N}_{\mathsf{K}}(x) = \left\{p \in \mathbb{R}^n : \langle p, y - x\rangle \leq 0 \text{ for all } y \in \mathsf{K}\right\}.$$

Given a representation of K by inequality constraints, $\mathsf{K} = \{x \in \mathbb{R}^n : Ax \leq b\}$ for $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^n$, it holds

$$\mathsf{N}_{\mathsf{K}}(x) = \{A^T y : y \geq 0 \text{ and } y^T(Ax - b) = 0\}.$$

The KKT conditions yield

$$\nabla\Phi(y^*) = \nabla\Phi(x) - c - \lambda^*, \tag{14}$$

where $\lambda^* \in \mathsf{N}_{\mathsf{K}}(y^*)$. Since $\mathsf{N}_{\mathsf{K}}(y^*) = \mathsf{N}_{\mathsf{K}_0}(y^*) + \mathsf{N}_{\mathbb{R}^n_+}(y^*)$, we can can decompose $\lambda^* = \beta - \alpha$ with $\beta \in \mathsf{N}_{\mathsf{K}_0}(y^*)$ and $-\alpha \in \mathsf{N}_{\mathbb{R}^n_+}(y^*)$. In particular, we have $\alpha \geq 0$ and $\alpha_i > 0 \implies y_i^* = 0$ for every $i = 1, \ldots, n$.

Substituting this into (13) gives

$$\begin{aligned}\mathsf{D}_\Phi(z \,\|\, y^*) - \mathsf{D}_\Phi(z \,\|\, x) &= -\Phi(y^*) + \Phi(x) + \langle \nabla\Phi(x), y^* - x\rangle + \langle c - \alpha + \beta, z - y^*\rangle \\ &\leq -\mathsf{D}_\Phi(y^* \,\|\, x) + \langle c - \alpha, z - y^*\rangle,\end{aligned}$$

where the inequality comes from $\langle \beta, z - y^*\rangle \leq 0$ since $z \in \mathsf{K}$ and $\beta \in \mathsf{N}_{\mathsf{K}}(y^*)$. We have proved the following.

Lemma 5 *For any $x, z \in \mathsf{K}$, and $c \in \mathbb{R}^n$, let $y^* = \Pi^c_{\mathsf{K}}(x)$ and λ^* be as in (14). Then for any $\alpha \in -\mathsf{N}_{\mathbb{R}^+_n}(y^*)$ such that $\lambda^* + \alpha \in \mathsf{N}_{\mathsf{K}_0}(y^*)$, it holds that*

$$\mathsf{D}_\Phi(z \,\|\, y^*) - \mathsf{D}_\Phi(z \,\|\, x) \leq \langle c - \alpha, z - y^*\rangle.$$

2.1. Iterative Bregman projections

We describe now a discretization of the algorithm from the introduction. Fix a tree T and recall the definition of K_T from (3). Let Q_T denote the collection of vectors $q \in \mathbb{R}_+^{V \setminus \{\mathbb{r}\}}$ such that for all $u \in V \setminus \mathcal{L}$,

$$\sum_{v \in \chi(u)} q_v = 1.$$

For $q \in Q_T$ and $u \in V \setminus \mathcal{L}$, we use $q^{(u)} \in \mathbb{R}_+^{\chi(u)}$ to denote the vector defined by $q_v^{(u)} := q_v$ for $v \in \chi(u)$, and define the corresponding probability simplex $Q_T^{(u)} := \{q^{(u)} : q \in Q_T\}$. We will use $\Delta : Q_T \to \mathsf{K}_T$ for the map which sends $q \in Q_T$ to the (unique) $x = \Delta(q) \in \mathsf{K}_T$ such that

$$x_v = x_u q_v \qquad \forall u \in V \setminus \mathcal{L}, v \in \chi(u).$$

Note that q contains more information than x; the map Δ fails to be invertible whenever there is some $u \in V \setminus \mathcal{L}$ with $x_u = 0$.

Fix $\kappa \geq 1$. Let $\mathcal{D}^{(u)}$ be the open $(\min_{v \in \chi(u)} \delta_v)$-neighborhood of $\mathbb{R}_+^{\chi(u)}$, and define the strictly convex function $\Phi^{(u)} : \mathcal{D}^{(u)} \to \mathbb{R}$ by

$$\Phi^{(u)}(p) := \frac{1}{\kappa} \sum_{v \in \chi(u)} \frac{w_v}{\eta_v} \left(p_v + \delta_v\right) \log\left(p_v + \delta_v\right).$$

Denote the corresponding Bregman divergence on $Q_T^{(u)}$ by

$$\mathsf{D}^{(u)}(p \,\|\, p') = \frac{1}{\kappa} \sum_{v \in \chi(u)} \frac{w_v}{\eta_v} \left[\left(p_v + \delta_v\right) \log \frac{p_v + \delta_v}{p'_v + \delta_v} + p'_v - p_v\right].$$

We now define an algorithm that takes a point $q \in Q_T$ and a cost vector $c \in \mathbb{R}_+^{\mathcal{L}}$ and outputs a point $p = \mathcal{A}(q, c) \in Q_T$. Fix $\langle u_1, u_2, \ldots, u_N\rangle$ a topological ordering of $V \setminus \mathcal{L}$ such that every child in T occurs before its parent. We define p inductively as follows. Let $\hat{c}_\ell := c_\ell$ for $\ell \in \mathcal{L}$. For every $j = 1, 2, \ldots, N$:

$$\hat{c}_v^{(u_j)} := \hat{c}_v \qquad \forall v \in \chi(u_j) \tag{15}$$

$$p^{(u_j)} := \operatorname{argmin}\left\{ \mathsf{D}^{(u_j)}\left(p \,\|\, q^{(u_j)}\right) + \left\langle p, \hat{c}^{(u_j)} \right\rangle \;\middle|\; p \in Q_T^{(u_j)} \right\} \tag{16}$$

$$\hat{c}_{u_j} := \sum_{v \in \chi(u_j)} p_v^{(u_j)}\, \hat{c}_v \tag{17}$$

Notice that (16) can be computed efficiently. In the remainder of this paper, we will sketch a proof how this algorithm can be used to obtain the guarantees of Theorem 1 and Theorem 3. Due to space constraints, most lemmas are only proved in the full version of our paper.

2.2. The global divergence

For $z \in \mathsf{K}_T$ and $q \in Q_T$, define the global divergence function

$$\tilde{\mathsf{D}}(z \,\|\, q) := \frac{1}{\kappa} \sum_{u \notin \mathcal{L}} \sum_{v \in \chi(u)} \frac{w_v}{\eta_v} \left[(z_v + \delta_v z_u) \log\left(\frac{\frac{z_v}{z_u} + \delta_v}{q_v + \delta_v} \right) + z_u q_v - z_v \right],$$

with the convention that $0 \log\left(\frac{0}{0} + \delta_v\right) = \lim_{\epsilon \to 0} \epsilon \log\left(\frac{0}{\epsilon} + \delta_v\right) = 0$. This is the Bregman divergence associated to (7) (divided by κ) with $\frac{x_v}{x_u}$ replaced by q_v. We will use $\tilde{\mathsf{D}}$ as a potential function to prove (1). The next lemma shows that when the offline algorithm moves, the change in potential is bounded by $O(1/\kappa)$ times the offline movement cost.

Lemma 6 *It holds that for any $q \in Q_T$ and $z, z' \in \mathsf{K}_T$,*

$$\left| \tilde{\mathsf{D}}(z \,\|\, q) - \tilde{\mathsf{D}}(z' \,\|\, q) \right| \le \frac{1}{\kappa}\left(2 + \frac{4}{\tau}\right) \|z - z'\|_{\ell_1(w)}\,.$$

We will sometimes implicitly restrict vectors $x \in \mathbb{R}^V$ to the subspace spanned by $\{e_\ell : \ell \in \mathcal{L}\}$. In this case, we employ the notation

$$\langle x, y \rangle_{\mathcal{L}} := \sum_{\ell \in \mathcal{L}} x_\ell y_\ell,$$

when either vector lies in $\mathbb{R}^V$ or $\mathbb{R}^{\mathcal{L}}$.

According to the following lemma, the change in potential due to movement of the online algorithm is bounded by the difference in service cost between the offline and online algorithm.

Lemma 7 *For any cost vector $c \in \mathbb{R}_+^{\mathcal{L}}$, $z \in \mathsf{K}_T$, and $q \in Q_T$, it holds that if $p = \mathcal{A}(q, c)$, then*

$$\tilde{\mathsf{D}}(z \,\|\, p) - \tilde{\mathsf{D}}(z \,\|\, q) \le \left\langle c, z - \Delta(p) \right\rangle_{\mathcal{L}}.$$

2.3. Algorithm and competitive analysis

For the proof of bound (2), we employ two potential functions ψ and Ψ, defined as follows. For $x \in \mathsf{K}_T$, let $\psi(x) := \sum_{u \neq \mathbb{r}} w_u x_u$. For $q \in Q_T$, let

$$\Psi_u(q) := -\Delta(q)_u\, \mathsf{D}^{(u)}\left(\theta^{(u)} \,\|\, q^{(u)}\right)$$
$$\Psi(q) := \sum_{u \notin \mathcal{L}} \Psi_u(q).$$

The next lemma justifies that when the algorithm moves from x to y, it suffices to bound the positive movement cost $\left\|(x-y)_+\right\|_{\ell_1(w)}$ rather than the actual movement cost $\|x-y\|_{\ell_1(w)}$.

Lemma 8 *For $x, y \in \mathsf{K}_T$ it holds that*

$$\|x-y\|_{\ell_1(w)} = 2\left\|(x-y)_+\right\|_{\ell_1(w)} + [\psi(y) - \psi(x)].$$

In the next section, we will prove the following.

Lemma 9 (Movement analysis) *It holds that*

$$\frac{\tau-3}{\kappa\tau}\left\|(x-y)_+\right\|_{\ell_1(w)} \le (2\mathfrak{D}_T + \log n)\langle c, x\rangle_{\mathcal{L}} + \left[\Psi(q) - \Psi(p)\right].$$

Define $w_{\min} := \min\{w_\ell : \ell \in \mathcal{L}\}$ and

$$\epsilon_T := \frac{w_{\min}}{2(2\mathfrak{D}_T + \log n)} \frac{\tau-3}{\tau\kappa}.$$

Theorem 10 *Consider any $q \in Q_T$ and $c \in \mathbb{R}_+^{\mathcal{L}}$. If we define $p = \mathcal{A}(q, c)$, $x = \Delta(q)$, $y = \Delta(p)$, then for any $z \in \mathsf{K}_T$:*

$$\langle c, y\rangle_{\mathcal{L}} \le \langle c, z\rangle_{\mathcal{L}} + \left[\tilde{\mathsf{D}}(z \,\|\, q) - \tilde{\mathsf{D}}(z \,\|\, p)\right] \tag{18}$$

$$\kappa^{-1}\left\|x-y\right\|_{\ell_1(w)} \le [\psi(y) - \psi(x)] + \frac{2\tau}{\tau-3}\left([\Psi(q) - \Psi(p)] + (2\mathfrak{D}_T + \log n)\langle c, x\rangle_{\mathcal{L}}\right). \tag{19}$$

Moreover, if $\|c\|_\infty \le \epsilon_T$, then

$$\kappa^{-1}\left\|x-y\right\|_{\ell_1(w)} \le [\psi(y) - \psi(x)] + \frac{4\tau}{\tau-3}\left([\Psi(q) - \Psi(p)] + (2\mathfrak{D}_T + \log n)\langle c, y\rangle_{\mathcal{L}}\right). \tag{20}$$

Proof The bound (18) follows from Lemma 7, and (19) follows from Lemma 9 and Lemma 8. To see that (20) follows from (19) and Lemma 9, use the fact that

$$\langle c, x\rangle_{\mathcal{L}} \le \langle c, y\rangle_{\mathcal{L}} + \frac{\|c\|_\infty}{w_{\min}}\left\|(x-y)_+\right\|_{\ell_1(w)}.$$

■

In light of Theorem 10, we can respond to a cost function $c \in \mathbb{R}_+^{\mathcal{L}}$ by splitting it into M pieces $c_1, c_2, \ldots, c_M$ where $M = \lceil \|c\|_\infty / \epsilon_T \rceil$. Now define $q_i := \mathcal{A}(q_{i-1}, c/M)$, $q_0 := q$ and $\bar{\mathcal{A}}(q, c) := q_M$. From Theorem 10 and Lemma 6, we conclude the following theorem.

Theorem 11 *Fix $\tau \geq 4$. Consider the algorithm that begins in some configuration $q_0 \in Q_T$. If $c_t \in \mathbb{R}_+^{\mathcal{L}}$ is the cost function that arrives at time t, denote $q_t := \bar{\mathcal{A}}(q_{t-1}, c_t)$. Then the sequence $\langle \Delta(q_0), \Delta(q_1), \dots \rangle$ is an online algorithm that is $(1, O(1/\kappa), O(\kappa(\mathfrak{D}_T + \log n)))$-finely competitive.*

It is well-known (see Bansal et al. (2015)) that any n-point HST metric can be approximated by another HST metric induced by a tree of depth $O(\log n)$ while distorting distances by an $O(1)$ factor. In conjunction with the preceding theorem, this yields the validity of Theorem 1 and Theorem 3.

2.4. Movement analysis

It remains to prove Lemma 9. The KKT conditions (cf. (14)) give: For every $v \in \chi(u)$,

$$\frac{1}{\kappa}\frac{w_v}{\eta_v}\log\left(\frac{p_v + \delta_v}{q_v + \delta_v}\right) = \beta_u - \hat{c}_v + \alpha_v\,, \tag{21}$$

where $\alpha_v \geq 0$ is the Lagrange multiplier corresponding to the non-negativity constraint of q_v, and $\beta_u \geq 0$ is the multiplier corresponding to the constraint $\sum_{v \in \chi(u)} q_v \geq 1$. There is no multiplier for the constraint $\sum_{v \in \chi(u_j)} q_v^{(u_j)} \leq 1$ because this constraint will be satisfied automatically and is therefore not needed in (16): If it were violated, decreasing some p_v with $p_v > q_v^{(u_j)}$ would yield a strictly better solution to the minimization problem (16).

Lemma 12 *It holds that $\alpha_v \leq \hat{c}_v$ for all $v \in V \setminus \{\mathbb{r}\}$.*

Define $\sigma_v := \log\left(\frac{p_v+\delta_v}{q_v+\delta_v}\right)$ so that

$$q_v - p_v = (q_v + \delta_v)(1 - e^{\sigma_v}). \tag{22}$$

Recall that for $v \in \chi(u)$, we have $x_v = q_v x_u$ and $y_v = p_v y_u$, thus

$$x_v - y_v = x_u(q_v - p_v) + p_v(x_u - y_u) = (x_v + \delta_v x_u)(1 - e^{\sigma_v}) + p_v(x_u - y_u).$$

In particular,

$$\begin{aligned} w_v\left(x_v - y_v\right)_+ &\leq w_v(x_v + \delta_v x_u)(1 - e^{\sigma_v})_+ + w_v p_v\left(x_u - y_u\right)_+ \\ &\leq w_v(x_v + \delta_v x_u)(1 - e^{\sigma_v})_+ + \frac{w_u}{\tau} p_v\left(x_u - y_u\right)_+ . \end{aligned}$$

Using $\sum_{v \in \chi(u)} p_v = 1$ and summing over all vertices yields

$$\sum_{v \neq \mathbb{r}} w_v\left(x_v - y_v\right)_+ \leq \sum_{v \neq \mathbb{r}} w_v(x_v + \delta_v x_{\mathsf{p}(v)})(1 - e^{\sigma_v})_+ + \frac{1}{\tau}\sum_{v \neq \mathbb{r}} w_v\left(x_v - y_v\right)_+ ,$$

hence

$$\begin{aligned} \sum_{v \neq \mathbb{r}} w_v\left(x_v - y_v\right)_+ &\leq \frac{\tau}{\tau - 1}\sum_{v \neq \mathbb{r}} w_v(x_v + \delta_v x_{\mathsf{p}(v)})(1 - e^{\sigma_v})_+ \\ &\leq \frac{\tau}{\tau - 1}\sum_{v \neq \mathbb{r}} w_v(x_v + \delta_v x_{\mathsf{p}(v)})\,(\sigma_v)_- \\ &\leq \frac{\kappa\tau}{\tau - 1}\left(\sum_{v \neq \mathbb{r}} \eta_v x_v \hat{c}_v + \sum_{u \notin \mathcal{L}} x_u \sum_{v \in \chi(u)} \theta_v(\hat{c}_v - \alpha_v)\right), \end{aligned} \tag{23}$$

where the last line uses Lemma 12 and (21), to bound $w_v(\sigma_v)_- \le \kappa\eta_v(\hat{c}_v - \alpha_v)$.

Note that

$$\sum_{v \neq \mathbb{r}} \eta_v x_v \hat{c}_v \le \sum_{\ell \in \mathcal{L}} c_\ell x_\ell \sum_{v \in \gamma_{\mathbb{r},\ell} \setminus \{\mathbb{r}\}} \eta_v \le (\mathfrak{D}_T + \log n)\langle c, x\rangle, \tag{24}$$

since for any $\ell \in \mathcal{L}$, it holds that

$$\sum_{v \in \gamma_{\mathbb{r},\ell} \setminus \{\mathbb{r}\}} \eta_v = \mathfrak{D}_T(\ell) + \sum_{v \in \gamma_{\mathbb{r},\ell} \setminus \{\mathbb{r}\}} \log \frac{|\mathcal{L}_{\mathsf{p}(v)}|}{|\mathcal{L}_v|} = \mathfrak{D}_T(\ell) + \log n,$$

where $\mathfrak{D}_T(\ell)$ is the combinatorial depth of ℓ.

The second sum in (23) can be interpreted as the service cost of hybrid configurations of q and θ: While $\sum_{v \in \chi(u)} x_v \hat{c}_v$ is the service cost of x in $\mathcal{L}_u$, the term $x_u \sum_{v \in \chi(u)} \theta_v \hat{c}_v$ is the service cost in $\mathcal{L}_u$ of the modification of x whose conditional probabilities at the children of u are given by $\theta^{(u)}$ rather than $q^{(u)}$. To bound this hybrid service cost, we will employ the auxiliary potential Ψ.

2.4.1. THE HYBRID COST

The next lemma is crucial: It relates the service cost (with respect to the reduced cost $\hat{c} - \alpha$) of the hybrid configurations to the service cost of the actual configuration and the movement cost.

Lemma 13 *For any $u \notin \mathcal{L}$, it holds that*

$$\Psi_u(p) - \Psi_u(q) \le \frac{2}{\kappa}\frac{w_u}{\tau}\left(x_u - y_u\right)_+ + \sum_{v \in \chi(u)} (\hat{c}_v - \alpha_v)\left[x_v - \theta_v x_u\right]. \tag{25}$$

Using the lemma gives

$$\begin{aligned}\sum_{u \notin \mathcal{L}} x_u \sum_{v \in \chi(u)} \theta_v(\hat{c}_v - \alpha_v) &\overset{(25)}{\le} [\Psi(q) - \Psi(p)] + \frac{2}{\kappa\tau}\left\|\left(\Delta(q) - \Delta(p)\right)_+\right\|_{\ell_1(w)} + \sum_{v \neq \mathbb{r}} \hat{c}_v x_v \\ &\le [\Psi(q) - \Psi(p)] + \frac{2}{\kappa\tau}\left\|\left(\Delta(q) - \Delta(p)\right)_+\right\|_{\ell_1(w)} + \mathfrak{D}_T\langle c, x\rangle_{\mathcal{L}}.\end{aligned}$$

Combining this inequality with (23) and (24) gives

$$\kappa^{-1}\left\|(x - y)_+\right\|_{\ell_1(w)} \le \frac{\tau}{\tau - 1}\left[(2\mathfrak{D}_T + \log n)\langle c, x\rangle_{\mathcal{L}} + \left(\Psi(q) - \Psi(p)\right) + \frac{2}{\kappa\tau}\left\|(x - y)_+\right\|_{\ell_1(w)}\right],$$

completing the verification of Lemma 9.

Acknowledgments

The first author is partially funded by EPSRC Award 1652110. Most of the work was done while the first author was visiting University of Washington, hosted by the second author.

References

Jacob Abernethy, Peter Bartlett, Niv Buchbinder, and Isabelle Stanton. A regularization approach to metrical task systems. In *Algorithmic Learning Theory*, ALT 2010. Springer, 2010.

Nikhil Bansal, Niv Buchbinder, and Joseph Naor. Metrical task systems and the k-server problem on HSTs. In *Proceedings of the 37th International Colloquium Conference on Automata, Languages and Programming*, ICALP'10, pages 287–298, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-14164-1, 978-3-642-14164-5. URL http://dl.acm.org/citation.cfm?id=1880918.1880950.

Nikhil Bansal, Niv Buchbinder, and Joseph Naor. A primal-dual randomized algorithm for weighted paging. *J. ACM*, 59(4):Art. 19, 24, 2012. ISSN 0004-5411. doi: 10.1145/2339123.2339126. URL http://dx.doi.org/10.1145/2339123.2339126.

Nikhil Bansal, Niv Buchbinder, Aleksander Madry, and Joseph Naor. A polylogarithmic-competitive algorithm for the k-server problem. *J. ACM*, 62(5):Art. 40, 49, 2015. ISSN 0004-5411. doi: 10.1145/2783434. URL http://dx.doi.org/10.1145/2783434.

Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193, 1996. doi: 10.1109/SFCS.1996.548477. URL https://doi.org/10.1109/SFCS.1996.548477.

Yair Bartal, Avrim Blum, Carl Burch, and Andrew Tomkins. A polylog(n)-competitive algorithm for metrical task systems. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 711–719, New York, NY, USA, 1997. ACM. ISBN 0-89791-888-6. doi: 10.1145/258533.258667. URL http://doi.acm.org/10.1145/258533.258667.

Yair Bartal, Nathan Linial, Manor Mendel, and Assaf Naor. On metric Ramsey-type phenomena. *Ann. of Math. (2)*, 162(2):643–709, 2005. ISSN 0003-486X. doi: 10.4007/annals.2005.162.643. URL http://dx.doi.org/10.4007/annals.2005.162.643.

Yair Bartal, Béla Bollobás, and Manor Mendel. Ramsey-type theorems for metric spaces with applications to online problems. *J. Comput. System Sci.*, 72(5):890–921, 2006. ISSN 0022-0000. doi: 10.1016/j.jcss.2005.05.008. URL http://dx.doi.org/10.1016/j.jcss.2005.05.008.

Avrim Blum, Howard Karloff, Yuval Rabani, and Michael Saks. A decomposition theorem for task systems and bounds for randomized server problems. *SIAM J. Comput.*, 30(5):1624–1661, 2000. ISSN 0097-5397. URL https://doi-org.offcampus.lib.washington.edu/10.1137/S0097539799351882.

Allan Borodin, Nathan Linial, and Michael E. Saks. An optimal on-line algorithm for metrical task system. *J. ACM*, 39(4):745–763, October 1992. ISSN 0004-5411. doi: 10.1145/146585.146588. URL http://doi.acm.org/10.1145/146585.146588.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012. doi: 10.1561/2200000024. URL https://doi.org/10.1561/2200000024.

Sébastien Bubeck, Michael B. Cohen, James R. Lee, and Yin-Tat Lee. Metrical task systems on trees via mirror descent and unfair gluing. 2018a.

Sébastien Bubeck, Michael B. Cohen, Yin Tat Lee, James R. Lee, and Aleksander Madry. k-server via multiscale entropic regularization. In *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, 2018b. doi: 10.1145/3188745.3188798. URL http://doi.acm.org/10.1145/3188745.3188798.

Niv Buchbinder and Joseph Naor. The design of competitive online algorithms via a primal-dual approach. *Found. Trends Theor. Comput. Sci.*, 3(2-3):front matter, 93–263 (2009), 2007. ISSN 1551-305X. doi: 10.1561/0400000024.

Niv Buchbinder, Shahar Chen, and Joseph (Seffi) Naor. Competitive analysis via regularization. In *Proceedings of the Twenty-fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 436–444, Philadelphia, PA, USA, 2014. Society for Industrial and Applied Mathematics.

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.

Amos Fiat and Manor Mendel. Better algorithms for unfair metrical task systems and applications. *SIAM Journal on Computing*, 32(6):1403–1422, 2003. doi: 10.1137/S0097539700376159. URL https://doi.org/10.1137/S0097539700376159.

Steve Seiden. Unfair problems and randomized algorithms for metrical task systems. *Inf. Comput.*, 148(2):219–240, February 1999. ISSN 0890-5401. doi: 10.1006/inco.1998.2744. URL http://dx.doi.org/10.1006/inco.1998.2744.