Optimal robustness-consistency tradeoffs for learning-augmented metrical task systems

Nicolas Christianson **Junxuan Shen** **Adam Wierman**
California Institute of Technology

Abstract

We examine the problem of designing *learning-augmented* algorithms for *metrical task systems* (MTS) that exploit machine-learned advice while maintaining rigorous, worst-case guarantees on performance. We propose an algorithm, DART, that achieves this dual objective, providing cost within a multiplicative factor $(1 + \epsilon)$ of the machine-learned advice (i.e., *consistency*) while ensuring cost within a multiplicative factor $2^{\mathcal{O}(1/\epsilon)}$ of a baseline robust algorithm (i.e., *robustness*) for any $\epsilon > 0$. We show that this exponential tradeoff between consistency and robustness is unavoidable in general, but that in important subclasses of MTS, such as when the metric space has bounded diameter and in the k-server problem, our algorithm achieves improved, polynomial tradeoffs between consistency and robustness.

1 INTRODUCTION

The *metrical task systems* (MTS) problem is a central problem in the theory of online algorithms, encompassing a wide range of problems broadly characterized as "online optimization with switching costs" such as *convex function chasing* (CFC) and k-*server*. In MTS, a decision-maker is faced with a metric space (X, d) and a sequence of adversarial cost functions $f_1, \dots, f_T : X \to [0, +\infty]$ that are revealed *online*; after the function f_t is revealed, the decision-maker chooses a decision $x_t \in X$ and pays the *service* cost $f_t(x_t)$ as well as the *switching* or *movement* cost $d(x_t, x_{t-1})$, which penalizes changing decisions. The MTS problem has deep connections with online learning (Blum and Burch, 1997; Buchbinder et al., 2012; Daniely and Mansour, 2019) and broad applicability to problems such as datacenter operation (Lin et al., 2012, 2013; Albers and Quedenfeld, 2021), smoothed online regression and clustering (Goel and Wierman, 2019; Bubeck et al., 2021b; Deng et al., 2022), and planning/logistics (Dehghani et al., 2017). MTS algorithms are designed to minimize the *competitive ratio*, which quantifies the worst-case ratio in cost between an algorithm and the offline optimal sequence of decisions (Definition 2.1). The competitive ratio of MTS algorithms grows in the cardinality or dimension of the decision space; for instance, if $|X| = n$, any deterministic algorithm is $\Omega(n)$-competitive and any randomized algorithm is $\Omega(\log n)$-competitive (Borodin et al., 1992; Bubeck et al., 2022).

Due to the worst-case nature of the competitive ratio, traditional algorithms for MTS are conservative and may perform poorly in high-dimensional settings. In many real-world sequential decision-making tasks, however, significant data is available concerning typical problem instances, enabling data-driven, machine-learned (ML) algorithms to outperform traditional algorithms, which ignore such data. Despite this excellent practical performance, ML algorithms come with no *a priori* guarantees on worst-case behavior. As such, their performance may be jeopardized at deployment time if they are faced with distribution shift or unseen problem instances.

The tension between ML algorithms' excellent average-case performance and their lack of worst-case guarantees has motivated the development of *learning-augmented* algorithms for a wide range of online problems such as ski-rental, scheduling, and caching (Mahdian et al., 2012; Purohit et al., 2018; Lykouris and Vassilvtiskii, 2018; Mitzenmacher and Vassilvitskii, 2021). These algorithms are designed to exploit the performance of untrusted *advice* (e.g., from an ML algorithm) while maintaining rigorous guarantees on worst-case performance. Specifically, learning-augmented algorithms are designed to give simultaneous guarantees of *consistency* – a competitive guarantee against the advice – along with *robustness* – a worst-case competitive ratio guarantee (Definition 2.2). Tunable guarantees are typically sought so that $(1 + \epsilon)$-consistency can be obtained alongside bounded robustness for *any* $\epsilon > 0$, enabling better exploitation of good advice when ϵ is

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

chosen to be small.

Antoniadis et al. (2020) propose two algorithms that switch between an advice algorithm and a C-competitive algorithm for MTS or a special case thereof, giving guarantees of robustness and consistency for any MTS problem. In particular, their deterministic algorithm achieves 9-consistency and $9C$-robustness and their randomized algorithm achieves expected cost bounded by $\min\{(1+\epsilon)\mathrm{C}_{\mathrm{ADV}}, (1+\epsilon)C \cdot \mathrm{C}_{\mathrm{OPT}}\} + \mathcal{O}(\frac{D}{\epsilon})$, where $D = \mathrm{diam}(X)$ and $\mathrm{C}_{\mathrm{ADV}}, \mathrm{C}_{\mathrm{OPT}}$ are the advice and offline optimal costs, respectively. However, their deterministic algorithm cannot improve upon 9-consistency and the randomized algorithm is limited by the additive $\mathcal{O}(\frac{D}{\epsilon})$ term, which precludes obtaining arbitrarily small consistency (e.g., when $\mathrm{C}_{\mathrm{ADV}} = \mathcal{O}(1)$) and causes the bound to degrade or fail as the diameter of the metric space grows. This diameter-dependence is of particular limitation to special cases of MTS such as CFC and k-server, where the natural setting is an unbounded metric space like $\mathbb{R}^n$.

Several subsequent works obtain robustness and consistency bounds independent of diameter in special cases. Rutten et al. (2022) propose an algorithm achieving $(1+\epsilon)$-consistency and $2^{\tilde{\mathcal{O}}(\frac{1}{\alpha\epsilon})}$-robustness under certain conditions on $\alpha, \epsilon > 0$ when service cost functions f_t are restricted to be *α-polyhedral* (Definition 2.4). In the case of convex function chasing (Section 2.3) on $(\mathbb{R}^n, \|\cdot\|_{\ell^2})$, Christianson et al. (2022) propose a $(\sqrt{2}+\epsilon)$-consistent, $\mathcal{O}(\frac{n}{\epsilon^2})$-robust algorithm, and Rutten et al. (2022) give a $(1+\epsilon)$-consistent, $\mathcal{O}(\frac{1}{\epsilon^2})$-robust algorithm for the one-dimensional case ($n = 1$). Lindermayr et al. (2022) give an algorithm for k-server (Section 2.3) on $\mathbb{R}$ that achieves $(1+\mathcal{O}(\epsilon))$-consistency and $\mathcal{O}(\frac{1}{\epsilon^{k-1}})$-robustness.

These latter results indicate that in certain subclasses of MTS, it is possible to obtain robustness and consistency bounds that are independent of metric space diameter. However, these results only exist for a few subclasses of MTS, and do not always guarantee $(1+\epsilon)$-consistency for arbitrarily small $\epsilon > 0$, thus limiting the exploitation of good advice. The following, important question remains open: *Does there exist a general algorithm for* MTS *and its subclasses that achieves $(1+\epsilon)$-consistency for any $\epsilon > 0$ while simultaneously maintaining robustness bounded independently of the metric space diameter?*

Contributions. In this work, we answer the above question in the affirmative. Specifically, we propose a randomized algorithm, DART (Algorithm 1), that, given any advice and any C-competitive algorithm for MTS or a special case thereof, achieves $(1+\epsilon)$-consistency and $2^{\mathcal{O}(1/\epsilon)}C$-robustness (Theorem 3.1), with robustness independent of the diameter of the metric space.

Our main result implies several robustness and consistency bounds for subclasses of MTS (Corollary 3.1.1), which we summarize in Table 1. In particular, we answer the question posed by Christianson et al. (2022) of whether $(1+\epsilon)$-consistency and bounded robustness can be achieved for convex function chasing (CFC) on unbounded domains with ϵ arbitrarily close to 0. We further prove lower bounds on robustness and consistency for MTS and CFC, showing that our upper bounds are essentially tight: *any* $(1+\epsilon)$-consistent algorithm must have robustness $2^{\Omega(1/\epsilon)}$ (Theorems 4.1, 4.2). Despite this exponential tradeoff for MTS and CFC in general settings, we show by a refined analysis that DART actually achieves robustness $\mathcal{O}(\frac{1}{\epsilon})$ when the space's diameter is bounded, with an additive term matching the dependence on diameter of Antoniadis et al. (2020) (Theorem 5.1). Moreover, we find that DART achieves $\mathcal{O}(\frac{k}{\epsilon})$-robustness for the *$k$-server* problem, giving the best known robustness and consistency tradeoff in general metric spaces for this widely-studied special case of MTS (Theorem 5.3). Finally, we consider the problem of k-chasing convex, α-polyhedral functions, a generalization of both k-server and CFC, and we find that DART guarantees robustness $\mathcal{O}(\frac{k}{\alpha\epsilon})$ in the one-dimensional setting (Theorem 5.4).

Our algorithm, DART, is distinguished from prior learning-augmented algorithms for MTS in both its generality and its specific MTS-oriented design. Prior algorithms were either devised for different online problems and simply applied off-the-shelf to MTS with advice (e.g., the cow path and multiplicative weights algorithms of Antoniadis et al. (2020)), or heavily leveraged geometric and structural assumptions on the problem setting (e.g., convexity in Christianson et al. (2022), α-polyhedrality in Rutten et al. (2022), $X = \mathbb{R}$ in Lindermayr et al. (2022)). In contrast, DART works for any MTS or special case and is designed principally to achieve $(1+\epsilon)$-consistency with respect to an advice algorithm. Specifically, it operates by updating probabilities assigned to the advice and to a chosen competitive algorithm based on the costs incurred by each algorithm *as well as* the distance between the two algorithms' decisions. The dependence on this latter quantity is important, as this enables obtaining consistency and robustness independent of diameter, and the randomized algorithm of Antoniadis et al. (2020) lacks such a dependence.

Proving these results requires several technical contributions. DART's robustness is obtained by directly bounding the extent to which DART can be led astray by bad advice; this approach requires proving a lower bound on a broad class of sums that includes as a special case the harmonic series (Supplemental Section C.1). Moreover, the extremal case of this lower bound naturally leads to the robustness and consistency lower bound for MTS (Theorem 4.1). Furthermore, our lower bound on robustness and consistency for CFC (Theorem 4.2) makes novel use of an observation (due to Bubeck et al. (2021a)) that MTS instances on trees are equivalent, in a certain sense, to CFC instances in a weighted ℓ^1 space. To our knowledge, no prior work has

Table 1: Summary of prior results, with our contributions in bold. Recall $D = \operatorname{diam}(X)$.

Problem	Reference	Consistency	Robustness	Assumptions
MTS $\lvert X\rvert = n$	Bubeck et al. (2021a)	—	$\mathcal{O}(\log^2 n)\mathrm{C}_{\text{Opt}}$	
	Antoniadis et al. (2020)	$9 \cdot \mathrm{C}_{\text{Adv}}$	$9 \cdot \mathcal{O}(\log^2 n)\mathrm{C}_{\text{Opt}}$	
	Antoniadis et al. (2020)	$(1+\epsilon)\mathrm{C}_{\text{Adv}} + \mathcal{O}(\frac{D}{\epsilon})$	$(1+\epsilon)\mathcal{O}(\log^2 n)\mathrm{C}_{\text{Opt}} + \mathcal{O}(\frac{D}{\epsilon})$	$D < \infty$
	This work	$\boldsymbol{(1+\epsilon)\mathrm{C}_{\text{Adv}}}$	$\boldsymbol{\min\{2^{\mathcal{O}(1/\epsilon)}\mathcal{O}(\log^2 n)\mathrm{C}_{\text{Opt}}, \mathcal{O}(\frac{\log^2 n}{\epsilon})\mathrm{C}_{\text{Opt}} + \mathcal{O}(\frac{D}{\epsilon})\}}$	
CFC $X \subseteq \mathbb{R}^n$	Sellke (2020)	—	$n \cdot \mathrm{C}_{\text{Opt}}$	
	Antoniadis et al. (2020)	$9 \cdot \mathrm{C}_{\text{Adv}}$	$9n \cdot \mathrm{C}_{\text{Opt}}$	
	Antoniadis et al. (2020)	$(1+\epsilon)\mathrm{C}_{\text{Adv}} + \mathcal{O}(\frac{D}{\epsilon})$	$(1+\epsilon)n \cdot \mathrm{C}_{\text{Opt}} + \mathcal{O}(\frac{D}{\epsilon})$	$D < \infty$
	Christianson et al. (2022)	$(\sqrt{2}+\epsilon)\mathrm{C}_{\text{Adv}}$	$\mathcal{O}\left(\frac{n}{\epsilon^2}\right)\mathrm{C}_{\text{Opt}}$	Euclidean
	This work	$\boldsymbol{(1+\epsilon)\mathrm{C}_{\text{Adv}}}$	$\boldsymbol{\min\{2^{\mathcal{O}(1/\epsilon)} n \cdot \mathrm{C}_{\text{Opt}}, \mathcal{O}(\frac{n}{\epsilon})\mathrm{C}_{\text{Opt}} + \mathcal{O}(\frac{D}{\epsilon})\}}$	
k-server	Koutsoupias and Papadimitriou (1995)	—	$(2k-1)\mathrm{C}_{\text{Opt}}$	
	Antoniadis et al. (2020)	$9 \cdot \mathrm{C}_{\text{Adv}}$	$9(2k-1)\mathrm{C}_{\text{Opt}}$	
	Antoniadis et al. (2020)	$(1+\epsilon)\mathrm{C}_{\text{Adv}} + \mathcal{O}(\frac{D}{\epsilon})$	$(1+\epsilon)(2k-1)\mathrm{C}_{\text{Opt}} + \mathcal{O}(\frac{D}{\epsilon})$	$D < \infty$
	Lindermayr et al. (2022)	$(1+\mathcal{O}(\epsilon))\mathrm{C}_{\text{Adv}}$	$\mathcal{O}(\frac{1}{\epsilon^{k-1}})\mathrm{C}_{\text{Opt}}$	$X = \mathbb{R}$
	This work	$\boldsymbol{(1+\epsilon)\mathrm{C}_{\text{Adv}}}$	$\boldsymbol{\mathcal{O}(\frac{k}{\epsilon})\mathrm{C}_{\text{Opt}}}$	
k-chasing convex, α-polyhedral functions $X \subseteq \mathbb{R}^n$	Bubeck et al. (2021b), **This work**	—	$\boldsymbol{\mathcal{O}(\frac{k}{\alpha})}\mathrm{C}_{\text{Opt}}$	
	Antoniadis et al. (2020)	$9 \cdot \mathrm{C}_{\text{Adv}}$	$9 \cdot \mathcal{O}(\frac{k}{\alpha})\mathrm{C}_{\text{Opt}}$	
	Antoniadis et al. (2020)	$(1+\epsilon)\mathrm{C}_{\text{Adv}} + \mathcal{O}(\frac{D}{\epsilon})$	$(1+\epsilon)\mathcal{O}(\frac{k}{\alpha})\mathrm{C}_{\text{Opt}} + \mathcal{O}(\frac{D}{\epsilon})$	$D < \infty$
	This work	$\boldsymbol{(1+\epsilon)\mathrm{C}_{\text{Adv}}}$	$\boldsymbol{\min\{2^{\mathcal{O}(1/\epsilon)}\mathcal{O}(\frac{k}{\alpha})\mathrm{C}_{\text{Opt}}, \mathcal{O}(\frac{k}{\alpha\epsilon})\mathrm{C}_{\text{Opt}} + \mathcal{O}(\frac{D}{\epsilon})\}}$	
	This work	$\boldsymbol{(1+\epsilon)\mathrm{C}_{\text{Adv}}}$	$\boldsymbol{\mathcal{O}(\frac{k}{\alpha\epsilon})\mathrm{C}_{\text{Opt}}}$	$X = \mathbb{R}$

used this correspondence to translate performance bounds on MTS algorithms to results for CFC. Finally, the results we obtain in Section 5 all follow via *refined analyses* of the DART algorithm, and in particular do not require modification of the algorithm or prior knowledge of, e.g., the diameter bound. Thus, DART is a *unified* algorithm that matches or improves upon the best known results on robustness and consistency for MTS and many of its subclasses.

Notation. $\overline{\mathbb{R}}_+$ denotes the nonnegative extended reals. We define $[n] \coloneqq \{1, \dots, n\}$ for $n \in \mathbb{N}$. Metric spaces (X, d) are assumed to be complete and separable. For a metric space X, $\operatorname{diam}(X) \coloneqq \sup_{x,y \in X} d(x, y)$. $\Delta(X)$ denotes the set of (Borel) probability measures on the metric space X; when X is finite with cardinality n, we identify this with the simplex Δ_n. For $x \in X$, δ_x is the Dirac measure supported at x. In asymptotic notation involving ϵ, the implied regime is $\epsilon \to 0$.

2 MODEL & PRELIMINARIES

This section introduces the metrical task systems problem and several subclasses that have received significant attention. We motivate these problems with applications to data science and multi-agent logistics, and introduce the notion of *learning-augmented* algorithms that enable breaking past pessimistic worst-case guarantees.

2.1 Metrical Task Systems

Let (X, d) be a metric space. In the *metrical task systems* (MTS) problem, at each time $t \in [T]$, a player beginning from some position $x_0 \in X$ observes an adversarially-chosen cost function $f_t : X \to \overline{\mathbb{R}}_+$ and must choose a state $x_t \in X$ to move to. The player then pays both the *service* cost $f_t(x_t)$ as well as the *movement* cost $d(x_t, x_{t-1})$. The time horizon T is unknown to the player *a priori*. An *instance* of MTS is characterized by a metric space (X, d), a starting position x_0, and the cost function sequence $f_1, \dots, f_T$.

A *deterministic online algorithm* ALG for MTS is a sequence of maps $\mathrm{ALG}_t : (\overline{\mathbb{R}}_+^X)^t \to X$ which map the cost functions observed through time t to a decision in X for each $t \in [T]$. That is, upon observing the cost function f_t, $\mathrm{ALG}_t(f_1, \dots, f_t) \in X$ is the decision produced by ALG at time t. When the instance is implicitly understood, we suppress the arguments and simply write ALG_t for ALG's decision at time t. ALG thus incurs cost

$$\mathrm{C}_{\text{Alg}} \coloneqq \sum_{t=1}^{T} f_t(\mathrm{ALG}_t) + d(\mathrm{ALG}_t, \mathrm{ALG}_{t-1}).$$

We define the notation $\mathrm{C}_{\text{Alg}}(t, t') \coloneqq \sum_{\tau=t}^{t'} f_t(\mathrm{ALG}_\tau) + d(\mathrm{ALG}_\tau, \mathrm{ALG}_{\tau-1})$ to reflect ALG's total cost incurred from time t through t'; if $t > t'$, then $\mathrm{C}_{\text{Alg}}(t, t') \coloneqq 0$.

A *randomized* MTS algorithm produces its decisions randomly: $\text{ALG}_t \sim p_t \in \Delta(X)$. It suffices to describe a randomized algorithm by its marginal distribution over states at each time (see, e.g., Bubeck et al. (2021a)). That is, suppose ALG_t is distributed according to p_t at each time $t \in [T]$; then the least-cost way for ALG to move from p_{t-1} to p_t is to couple the two distributions so as to minimize expected movement. Thus consecutive decisions should be distributed jointly according the optimal Wasserstein-1 transportation plan between p_{t-1} and p_t:

$$(\text{ALG}_t, \text{ALG}_{t-1}) \sim \gamma_t \coloneqq \underset{\gamma \in \Pi(p_t, p_{t-1})}{\arg\min} \mathbb{E}[d(x_t, x_{t-1})],$$

where $(x_t, x_{t-1}) \sim \gamma$ and $\Pi(\mu, \nu)$ is the set of distributions over X^2 with marginals μ and ν. If ALG couples consecutive decisions according to γ_t, then $\mathbb{E}[d(\text{ALG}_t, \text{ALG}_{t-1})] = \mathbb{W}^1_X(p_t, p_{t-1})$, the Wasserstein-1 distance between p_t and p_{t-1}.[1] Thus, $\mathbb{E}[\text{C}_{\text{ALG}}] \coloneqq \sum_{t=1}^T f_t(p_t) + \mathbb{W}^1_X(p_t, p_{t-1})$, where $f_t(p_t) \coloneqq \mathbb{E}_{x \sim p_t}[f_t(x)]$.

The *offline optimal* algorithm OPT for an MTS instance chooses the hindsight optimal sequence of decisions:

$$\text{C}_{\text{OPT}} \coloneqq \inf_{x_1, \ldots, x_T \in X} \sum_{t=1}^T f_t(x_t) + d(x_t, x_{t-1}).$$

Algorithms for MTS are typically judged by their *competitive ratio*, an adaptive measure of performance against OPT or any other algorithm.

Definition 2.1. *A deterministic algorithm* ALG *is c-**competitive with respect to** another algorithm* ALG$'$ *if, on any problem instance,* $\text{C}_{\text{ALG}} \leq c \cdot \text{C}_{\text{ALG}'} + b$, *where b is independent of the problem instance. If* ALG$'$ *is* OPT, *we simply say that* ALG *is c-competitive, or has **competitive ratio** c. If* ALG *or* ALG$'$ *are randomized, we replace costs with expected costs in the inequality.*

When service cost functions are arbitrary, algorithms for MTS can only be competitive on metric spaces with finite cardinality $|X| = n \in \mathbb{N}$. In this case, the work function algorithm achieves the optimal deterministic competitive ratio of $2n-1$ (Borodin et al., 1992). However, randomization can improve performance, with state-of-the-art algorithms achieving competitive ratio $\mathcal{O}(\log^2 n)$ (Bubeck et al., 2021a; Coester and Lee, 2019), which is tight for certain metric spaces (Bubeck et al., 2022).

2.2 Consistency, Robustness, and Bicompetitiveness

The competitive ratio quantifies worst-case performance of an online algorithm; its focus on the worst case thus biases algorithm design toward more conservative algorithms. Moreover, as just noted, the competitive ratio of MTS algorithms degrades as $|X|$ grows. In practical applications, however, data on typical problem instances is available, and thus data-driven machine-learned algorithms may significantly outperform traditional competitive algorithms. Since these machine-learned algorithms generally lack worst-case performance guarantees, we seek to design algorithms that exploit the good performance of a machine-learned *advice* algorithm (hereafter, ADV) while maintaining worst-case competitiveness. This motivates the following definitions.

Definition 2.2. *Let* ADV *be an advice algorithm. An algorithm* ALG *is c-**consistent** if it is c-competitive with respect to* ADV. ALG *is said to be r-**robust** if it is r-competitive, regardless of the performance of* ADV.

Thus, if ADV is a machine-learned algorithm and ALG is c-consistent and r-robust, then ALG achieves performance within a multiplicative factor c of the machine-learned advice while maintaining a worst-case competitive ratio. In this work, we design algorithms with tunable guarantees of robustness and consistency, i.e. that can achieve $(1+\epsilon)$-consistency for any $\epsilon > 0$ while keeping bounded robustness. We approach this by designing *bicompetitive* algorithms, defined as follows.

Definition 2.3. *Let* ALG, ALG$'$, ALG$''$ *be three algorithms.* ALG *is (c, r)-**bicompetitive with respect to** $(\text{ALG}', \text{ALG}'')$ if* ALG *is both c-competitive with respect to* ALG$'$ *and r-competitive with respect to* ALG$''$.

It follows that if ALG is (c, r)-bicompetitive with respect to algorithms (ADV, ROB) and ROB is b-competitive, then ALG is c-consistent and rb-robust. Thus to design robust and consistent MTS algorithms, it suffices to design bicompetitive algorithms. We detail prior robustness and consistency results for MTS in Table 1.

2.3 Special Cases of MTS

We now briefly describe some important special cases of MTS that are of particular relevance for applications to data science and multi-agent planning. Select bounds on competitive ratio, robustness, and consistency from prior work are detailed in Table 1.

Convex function chasing. The problem of convex function chasing (CFC), also known as "smoothed" online convex optimization, is an MTS in which the metric space is a finite-dimensional normed vector space and cost functions f_t are restricted to be convex. The best known algorithm for CFC in an arbitrary n-dimensional normed vector space achieves competitive ratio n and improved performance of $\mathcal{O}(\min\{n, \sqrt{n \log T}\})$ in the Euclidean setting (Sellke, 2020). On the other hand, any algorithm for CFC in $\mathbb{R}^n$ with the ℓ^p norm has competitive ratio

[1] Note that the optimal transportation plan γ_t exists by the assumption made throughout that X is complete and separable.

$\Omega(\max\{\sqrt{n}, n^{1-1/p}\})$ (Bubeck et al., 2019). It is straightforward to see by Jensen's inequality and convexity of norms that a c-competitive randomized algorithm for CFC can be derandomized by taking the expectation, yielding a c-competitive deterministic algorithm for CFC (Bansal et al., 2015b).

A number of special cases of CFC in which cost functions have additional structure have received attention in the literature. For example, the case where each f_t is the $\{0, +\infty\}$ indicator of a convex set $K_t \subseteq \mathbb{R}^n$ is known as *convex body chasing* and was first considered by Friedman and Linial (1993); the case of well-conditioned f_t was considered by Argue et al. (2020). The class of α-polyhedral functions has been widely studied as a special case in the CFC literature (Chen et al., 2018; Lin et al., 2020; Zhang et al., 2021) and is defined as follows.

Definition 2.4. *Fix $\alpha > 0$ and a normed vector space $(\mathbb{R}^n, \|\cdot\|)$. A function $g : \mathbb{R}^n \to \overline{\mathbb{R}}_+$ is α-**polyhedral** if it has a unique minimizer $\mathbf{v} \in \mathbb{R}^n$, and for all $\mathbf{x} \in \mathbb{R}^n$, $g(\mathbf{x}) \geq g(\mathbf{v}) + \alpha\|\mathbf{x} - \mathbf{v}\|$.*

A simply greedy algorithm obtains competitive ratio $\max\{1, \frac{2}{\alpha}\}$ for CFC with α-polyhedral service costs (Zhang et al., 2021), but better results can be obtained in the Euclidean setting (Lin, 2022).

***k*-server.** In the k-server problem, we control k agents ("servers") residing in the metric space X, and at each time t, we receive a request $r_t \in X$ and must move one of the servers to r_t, paying the distance traveled by the server we moved to meet the request. It is straightforward to see this is an MTS on the metric space $\binom{X}{k}$ (i.e., unordered k-tuples of states in X) endowed with the minimal matching distance inherited from the metric on X. The service cost f_t enforces that one of the servers is located at r_t, so for $\mathbf{x}_t := \{x_t^{(1)}, \ldots, x_t^{(k)}\} \in \binom{X}{k}$, $f_t(\mathbf{x}_t) = \infty \cdot \mathbb{1}_{r_t \notin \mathbf{x}_t}$. The (deterministic) work function algorithm is $(2k-1)$-competitive for k-server on any metric space (Koutsoupias and Papadimitriou, 1995), and no deterministic algorithm can achieve competitive ratio better than k (Manasse et al., 1988). Significant work has been done establishing tighter bounds on deterministic algorithms in particular metric spaces as well as sublinear bounds for randomized algorithms; see Koutsoupias (2009) for a survey and Bansal et al. (2015a); Bubeck et al. (2018) for recent results. For brevity, we only invoke the $\mathcal{O}(k)$-competitiveness of the work function algorithm in our work.

***k*-chasing convex functions.** The problem of k-chasing convex functions is a generalization of both k-server and CFC: the setting is taken to be a finite-dimensional vector space $(\mathbb{R}^n, \|\cdot\|)$, and we maintain a set of k servers $\mathbf{x}_t := \{\mathbf{x}_t^{(1)}, \ldots, \mathbf{x}_t^{(k)}\} \in \binom{\mathbb{R}^n}{k}$. At time t, an adversary serves a convex cost function $g_t : \mathbb{R}^n \to \overline{\mathbb{R}}_+$, and after moving our servers (by the triangle inequality, it suffices to just move one), we pay the service cost $\min_{i\in[k]} g_t(\mathbf{x}_t^{(i)})$ and the movement cost. Similar to k-server, this is an MTS on the metric space $\binom{\mathbb{R}^n}{k}$ endowed with the minimal matching distance inherited from the norm, with service costs f_t of the form $f_t(\mathbf{x}_t) := \min_{i\in[k]} g_t(\mathbf{x}_t^{(i)})$. This problem was introduced by Bubeck et al. (2021b), who found that under suitable structural assumptions on the functions g_t, competitive guarantees from existing k-server algorithms can be translated to k-chasing. However, they obtain randomized algorithms with guarantees dependent on adaptivity of the adversary. For the sake of clarity, in our work we consider k-chasing of convex, *α-polyhedral* functions (Definition 2.4), which enable translating deterministic algorithms for k-server into deterministic algorithms for k-chasing. In particular, following the proof of (Bubeck et al., 2021b, Theorem 3.1), we have the following result, which is proved in Supplementary Section B.

Proposition 2.5. *Let $g_1, \ldots, g_T : \mathbb{R}^n \to \overline{\mathbb{R}}_+$ be an instance of k-chasing convex, α-polyhedral functions. If* ALG *is a deterministic, C-competitive algorithm for k-server, then applying* ALG *to the sequence of minimizers $\mathbf{v}_1, \ldots, \mathbf{v}_T$ of $g_1, \ldots, g_T$ achieves competitive ratio at most $C \max\{1, \frac{2}{\alpha}\}$ for the k-chasing instance.*

It follows that the algorithm feeding the minimizers $v_1, \ldots, v_T$ to the $(2k-1)$-competitive work function algorithm as requests is $(2k-1)\max\{1, \frac{2}{\alpha}\}$-competitive for k-chasing convex α-polyhedral functions.

2.4 Example: Smoothed Online Clustering

Besides its deep connections to online learning (Blum and Burch, 1997; Buchbinder et al., 2012), MTS and its special cases have numerous applications to problems in data science: many online decision-making problems that penalize switching between decisions can be modeled within this framework. For instance, MTS has been applied to contextual Bayesian optimization with switching costs (Ramesh et al., 2022), CFC has applications to smoothed online regression (Goel and Wierman, 2019; Li et al., 2021) and k-server and its generalizations have applications to multiagent planning and logistics (Dehghani et al., 2017), dynamic clustering (Deng et al., 2022; Bubeck et al., 2021b), and beyond. To motivate our work designing learning-augmented algorithms for MTS, we briefly detail an application of k-chasing to smoothed online k-median clustering, highlighting the value delivered by utilizing machine-learned algorithms.

Suppose a decision-maker seeks to cluster a stream of points arriving online and must pay both for the distance between a point and the cluster center it is assigned to, as well as for the movement of cluster centers as they are updated to accommodate new arrivals. When the clustering objective is the k-median objective, this problem is naturally modeled as an instance of k-chasing convex, α-

Algorithm 1: $\textsc{Dart}(\textsc{Adv}, \textsc{Rob}; \epsilon)$

Input: Algorithms $\textsc{Adv}, \textsc{Rob}$; parameter $\epsilon > 0$
Output: Distributions $p_1, \ldots, p_T \in \Delta(X)$ chosen online

1 $\lambda_0 \leftarrow 0$
2 **for** $t = 1, 2, \ldots, T$ **do**
3 Observe f_t, $a_t \coloneqq \textsc{Adv}_t$, and $r_t \coloneqq \textsc{Rob}_t$
4 **if** $\mathrm{C}_{\textsc{Rob}}(1,t) \geq \frac{\epsilon}{4} \cdot \mathrm{C}_{\textsc{Adv}}(1,t)$ **then**
5 $\lambda_t \leftarrow 1$
6 **else**
7 $\lambda_t \leftarrow \max\left\{\lambda_{t-1} - \frac{\frac{\epsilon}{2}\mathrm{C}_{\textsc{Adv}}(t,t) + (1-\lambda_{t-1}) f_t(a_t)}{2d(a_{t-1}, r_{t-1})}, 0\right\}$
8 $p_t \leftarrow \lambda_t \delta_{a_t} + (1-\lambda_t)\delta_{r_t}$
9 **end**

polyhedral functions g_t of the form $g_t(\mathbf{x}) = c_t \|\mathbf{x} - \mathbf{v}_t\|_{\ell^1}$. The decision-maker's server positions $\mathbf{x}_t \in \binom{\mathbb{R}^n}{k}$ encode the k cluster centers, the minimizer $\mathbf{v}_t$ represents the position of the new arrival, and the service cost $f_t(\mathbf{x}_t) = \min_{i \in [k]} g_t(\mathbf{x}_t^{(i)})$ gives the cost of assigning the new arrival to the nearest cluster center. The weight c_t in the cost reflects the tradeoff between movement and service cost, i.e., the tradeoff between stability of the clustering and cost of adding new arrivals to the existing clusters.

When α is small and the number of clusters k is large, traditional competitive algorithms for k-chasing may perform poorly on this smoothed clustering objective, since the competitive ratio scales like $\frac{k}{\alpha}$ (Proposition 2.5). However, machine-learned algorithms trained on "typical" examples of evolving datasets may perform much better, motivating the development of algorithms that can be *consistent* with respect to such ML advice while maintaining worst-case guarantees on *robustness*.

3 A BICOMPETITIVE ALGORITHM FOR METRICAL TASK SYSTEMS

We now present a randomized algorithm, $\textsc{Dart}$ (**D**istance-**A**daptive **R**obust Weight **T**ransport, Algorithm 1), that achieves a bicompetitive guarantee $(1+\epsilon, 2^{\mathcal{O}(1/\epsilon)})$ in expectation with respect to *any* pair of (randomized) MTS algorithms $(\textsc{Adv}, \textsc{Rob})$.

The algorithm works as follows: it maintains a mixing weight $\lambda_t \in [0,1]$ associated with the decision $a_t \coloneqq \textsc{Adv}_t$ at each time t. This weight is adaptively updated at each timestep after observing the decisions made by $\textsc{Adv}$ and $\textsc{Rob}$, as well as their relative costs and the distance between the two algorithms' decisions (lines 4-7). $\textsc{Dart}$ then chooses its decision according to the distribution p_t (line 8), which takes value $\textsc{Adv}_t$ with probability λ_t and $\textsc{Rob}_t$ with probability $(1-\lambda_t)$. The parameter $\epsilon > 0$ provided as input to $\textsc{Dart}$ governs how closely $\textsc{Dart}$ follows $\textsc{Adv}$, i.e. how much we choose to "trust" the advice. A choice of ϵ that is very small will cause λ_t to stay closer to 1, giving better consistency in exchange for possibly worse robustness. On the other hand, a larger choice of ϵ will cause the weight λ_t to decrease more rapidly toward 0 in line 7, leading $\textsc{Dart}$ to more closely follow $\textsc{Rob}$ and improving worst-case robustness. As $\textsc{Dart}$ is a randomized algorithm, we couple consecutive distributions p_{t-1}, p_t according to the optimal (Wasserstein-1) transportation plan, as discussed in Section 2.1.

The following theorem explicitly characterizes the performance of $\textsc{Dart}$.

Theorem 3.1. *Let* $\textsc{Adv}, \textsc{Rob}$ *be any two (possibly randomized) algorithms for* MTS *or a special case thereof. For any chosen* $\epsilon > 0$*, Algorithm 1 (*$\textsc{Dart}$*) achieves bicompetitiveness* $\left(1+\epsilon, 2^{\mathcal{O}(1/\epsilon)}\right)$ *in expectation against* $(\textsc{Adv}, \textsc{Rob})$.

Our proof, which is presented in Supplementary Section C.1, consists of two parts. We first prove competitiveness with respect to $\textsc{Adv}$ via amortized analysis, using the potential function $\mathbb{E}_{x_t \sim p_t}[d(x_t, a_t)]$. We then prove competitiveness with respect to $\textsc{Rob}$ by means of a novel sum argument, upper bounding λ_t in terms of the cost incurred by $\textsc{Adv}$. This bound explicitly characterizes how much cost $\textsc{Dart}$ can be forced to incur by a "bad" advice algorithm $\textsc{Adv}$ before transferring all of its weight to $\textsc{Rob}$.

As immediate corollaries to Theorem 3.1, we obtain the following upper bounds on robustness and consistency for MTS, CFC, k-server, and k-chasing, which are proved in Supplementary Section C.2.

Corollary 3.1.1. *Choose any* $\epsilon > 0$.

i. *There is a* $(1+\epsilon)$*-consistent,* $2^{\mathcal{O}(1/\epsilon)}\mathcal{O}(\log^2(n))$*-robust randomized algorithm for* MTS *on any* n*-point metric space.*
ii. *There is a* $(1+\epsilon)$*-consistent,* $2^{\mathcal{O}(1/\epsilon)}n$*-robust deterministic algorithm for* CFC *on any* n*-dimensional normed vector space.*
iii. *There is a* $(1+\epsilon)$*-consistent,* $2^{\mathcal{O}(1/\epsilon)}(2k-1)$*-robust randomized algorithm for* k*-server on any metric space.*
iv. *There is a* $(1+\epsilon)$*-consistent,* $2^{\mathcal{O}(1/\epsilon)}\mathcal{O}(\frac{k}{\alpha})$*-robust randomized algorithm for* k*-chasing convex,* α*-polyhedral functions on any normed vector space.*

We wish to emphasize that the bicompetitive bound in Theorem 3.1 is the *first* bicompetitive bound for general MTS that is both independent of metric space diameter and provides bounded competitiveness with respect to $\textsc{Rob}$ for arbitrarily small $\epsilon > 0$. This latter property is of particular significance to practical application since this enables $\textsc{Dart}$ to achieve performance arbitrarily close to that of a black-box ML algorithm for MTS while maintaining a worst-case competitive guarantee, enabling better exploitation of the good ML performance.

In addition, the bound's independence from diameter enables obtaining robustness guarantees on unbounded spaces: Corollary 3.1.1.ii resolves the question of Christianson et al. (2022) of whether $(1+\epsilon)$-consistency and bounded robustness can be achieved for CFC on unbounded domains with arbitrary $\epsilon > 0$, and Corollaries 3.1.1.iii and iv answer for the first time the analogous question for k-server and k-chasing of convex, α-polyhedral functions. Although most practical problems have finite (but potentially very large) diameter, the robustness bounds given by DART still improve on the diameter-dependent results of Antoniadis et al. (2020) and Christianson et al. (2022) when $\operatorname{diam}(X) = 2^{\omega(1/\epsilon)}$. Moreover, as we will discuss in Section 5, DART achieves even better robustness when the diameter is bounded, matching the dependence of these other results and giving further-improved bounds for the k-server problem. To complement these theoretical advancements, we present in Supplementary Section A experimental results comparing DART against prior state-of-the-art algorithms for learning-augmented MTS.

4 FUNDAMENTAL LIMITS ON ROBUSTNESS AND CONSISTENCY

Though the tradeoff between robustness and consistency given by DART is exponential, it turns out that this is the best that we can hope for from any robust and consistent MTS algorithm. In the following theorem, which is proved in Supplementary Section D.1, we present a lower bound on the robustness of *any* $(1+\epsilon)$-consistent randomized MTS algorithm, showing that it must be exponential in $1/\epsilon$.

Theorem 4.1. *Let $\epsilon \in (0,1]$. There is an* MTS *instance on a finite metric space (X,d) with $|X| = \mathcal{O}(\frac{1}{\epsilon})$ and an adversarial advice algorithm* ADV *such that* any *randomized algorithm achieving $(1+\epsilon)$-consistency with respect to* ADV *is $2^{\Omega(1/\epsilon)}$-robust.*

Since the metric space X in the preceding theorem has cardinality $\mathcal{O}(\frac{1}{\epsilon})$, DART achieves robustness $2^{\mathcal{O}(1/\epsilon)}$, by Corollary 3.1.1.i. Thus, DART yields the *optimal* robustness-consistency tradeoff for general metrical task systems, up to constant factors in the exponent. Moreover, the metric space realizing the lower bound in Theorem 4.1 is not a pathological example: it is simply a finite subset of $\mathbb{R}$ with the usual (Euclidean) metric. Further note that this lower bound does not contradict the diameter-dependent upper bound of Antoniadis et al. (2020): the metric space has diameter exponential in $1/\epsilon$, and hence the randomized algorithm of Antoniadis et al. (2020) also obtains exponential robustness in this setting.

This exponential lower bound on the tradeoff between robustness and consistency for MTS raises the question of whether improved tradeoffs can be obtained for special cases of MTS where there is added structure. In particular, could the convexity inherent in CFC yield an improved dependence on ϵ in the robustness? In the following theorem, we answer this question in the negative, showing that in certain normed vector spaces the robustness-consistency tradeoff remains exponential.

Theorem 4.2. *Let $\epsilon \in (0,1]$. There is a* CFC *instance in $\mathbb{R}^{\mathcal{O}(1/\epsilon)}$ endowed with a weighted ℓ^1 norm, along with an adversarial advice algorithm* ADV*, such that* any *algorithm that is $(1+\epsilon)$-consistent with respect to* ADV *has robustness $2^{\Omega(1/\epsilon)}$.*

We present a proof in Supplementary Section D.2; it follows via a reduction to the MTS instance realizing the lower bound of Theorem 4.1, using the fact that MTS instances on a tree metric can be isometrically converted into CFC instances in a weighted ℓ^1 space (à la Bubeck et al. (2021a)). To the best of our knowledge, our use of this correspondence to obtain lower bounds on the performance of algorithms for CFC is novel.

As Corollary 3.1.1.ii gives a $(1+\epsilon)$-consistent, $2^{\mathcal{O}(1/\epsilon)}$-robust algorithm for CFC in a normed vector space of dimension $\mathcal{O}(\frac{1}{\epsilon})$, DART thus achieves the *optimal* tradeoff between robustness and consistency for CFC in general normed vector spaces, up to constant factors in the exponent. Note that this leaves open the question of whether subexponential robustness can be achieved for CFC under other norms such as the Euclidean norm.

5 BREAKING THE EXPONENTIAL ROBUSTNESS BARRIER

In Sections 3 and 4, we saw that DART achieves $(1+\epsilon, 2^{\mathcal{O}(1/\epsilon)})$-bicompetitiveness, and that the resultant tradeoff between robustness and consistency is optimal in general for MTS and CFC. However, prior work has obtained subexponential robustness bounds in certain special cases of MTS, including for CFC and k-server on the real line (Rutten et al., 2022; Lindermayr et al., 2022). In addition, for spaces with diameter bounded by some finite constant D, $(1+\epsilon)$-consistency and $\mathcal{O}(\frac{1}{\epsilon})$-robustness can be obtained for CFC in n dimensions with an additive term $\mathcal{O}(\frac{D}{\epsilon})$ on the robustness (Christianson et al., 2022), and a similar bound holds for MTS more generally (Antoniadis et al., 2020).

Given that Theorem 3.1 suggests an exponential bicompetitive tradeoff for DART, it is worth asking whether DART can perform better on such "easier" problem instances. It turns out that this is the case: we can prove that, in several special cases, DART achieves $(1+\epsilon)$-consistency together with robustness that depends only linearly on $\frac{1}{\epsilon}$. Notably, none of these improved bounds require modification of DART: they simply follow by a refined analysis. We consider three cases in turn.

Bounded diameter. When the metric space has bounded diameter D, and more generally when the algorithms $\textsc{Adv}$ and $\textsc{Rob}$ are never farther apart than a distance D, $\textsc{Dart}$ achieves bicompetitiveness $(1+\epsilon, \mathcal{O}(\frac{1}{\epsilon}))$ with respect to $(\textsc{Adv}, \textsc{Rob})$, with just an additive term of $\mathcal{O}(\frac{D}{\epsilon})$ on its competitiveness with respect to $\textsc{Rob}$. This matches the dependence on diameter obtained in prior work (Antoniadis et al., 2020; Christianson et al., 2022). Note that this bound does not require advance knowledge of the fact that the diameter is bounded: it simply results from a specialized analysis in the case that the algorithms $\textsc{Adv}$ and $\textsc{Rob}$ only take values in a subset of the metric space X that has diameter D. We present the formal performance bound in the following theorem.

Theorem 5.1. *Let* $\textsc{Adv}, \textsc{Rob}$ *be any two (possibly randomized) algorithms for* MTS *or a special case thereof. For any chosen* $\epsilon > 0$*, if* $d(\textsc{Adv}_t, \textsc{Rob}_t) \leq D$ *for all* $t \in [T]$*, Algorithm 1 (*$\textsc{Dart}$*) achieves cost bounded as*

$$\mathrm{C}_{\textsc{Dart}} \leq \min\left\{(1+\epsilon)\mathrm{C}_{\textsc{Adv}}, \mathcal{O}\left(\frac{1}{\epsilon}\right)\mathrm{C}_{\textsc{Rob}} + \mathcal{O}\left(\frac{D}{\epsilon}\right)\right\}.$$

That is, $\textsc{Dart}$ *is* $(1+\epsilon, \mathcal{O}(1/\epsilon))$*-bicompetitive against* $(\textsc{Adv}, \textsc{Rob})$*, with an additive constant* $\mathcal{O}(D/\epsilon)$ *on its competitive guarantee against* $\textsc{Rob}$.

This result is proved in Supplementary Section E.1. It is worth emphasizing that this bicompetitive guarantee holds *in addition* to the exponential tradeoff given by Theorem 3.1. Thus $\textsc{Dart}$ is $(1+\epsilon)$-competitive with respect to $\textsc{Adv}$ and has cost bounded by $\mathrm{C}_{\textsc{Rob}}$ as

$$\mathrm{C}_{\textsc{Dart}} \leq \min\left\{\mathcal{O}\left(\frac{1}{\epsilon}\right)\mathrm{C}_{\textsc{Rob}} + \mathcal{O}\left(\frac{D}{\epsilon}\right), 2^{\mathcal{O}(1/\epsilon)}\mathrm{C}_{\textsc{Rob}}\right\}.$$

As such, $\textsc{Dart}$ achieves the "best of both worlds" in terms of robustness, regardless of whether D is small or large. This simultaneous bound further extends to all the robustness bounds in Corollary 3.1.1.

***k*-server.** We next consider the k-server problem. In doing so, we restrict $\textsc{Adv}$ and $\textsc{Rob}$ to be *lazy* algorithms for k-server, i.e., algorithms that at any timestep move at most one server, moving none if the current server positions already satisfy the request. This assumption is without loss of generality (Borodin and El-Yaniv, 2005, §10.2.3). We also make the assumption that all servers begin at the same location; relaxing this assumption only changes the results by a constant additive term.

We first state a lemma relating the distance between any two lazy k-server algorithms to the offline optimal cost; the lemma is proved in Supplementary Section E.2.

Lemma 5.2. *Let* $s_1, \dots, s_T \in X$ *be the request sequence for a* k*-server instance on the metric space* (X, d)*, and let* $\textsc{Adv}$ *and* $\textsc{Rob}$ *be any two (possibly randomized)* k*-server algorithms. Further suppose that* $\textsc{Adv}$ *and* $\textsc{Rob}$ *are both lazy, and that all their servers start at the same point* $x_0 \in X$*. Let* $\mathbf{a}_1, \dots, \mathbf{a}_T \in \binom{X}{k}$ *and* $\mathbf{r}_1, \dots, \mathbf{r}_T \in \binom{X}{k}$ *be the sequences of server positions of* $\textsc{Adv}$ *and* $\textsc{Rob}$*, respectively for the problem instance. Then for any time* $t \in [T]$*,*

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \leq k \cdot \mathrm{C}_{\textsc{Opt}}(1, t),$$

where d_{mm} *is the minimal matching distance inherited from the metric* d*.*

Given any metric space (X, d), Lemma 5.2 allows us to bound the diameter of the subset of X that can be occupied by a lazy algorithm for a k-server instance by k times the offline optimal cost on that instance. Substituting this bound into Theorem 5.1 and using the fact that the work function algorithm is $(2k-1)$-competitive, we obtain the following result.

Theorem 5.3. *Consider* k*-server on an arbitrary metric space with all servers starting at some* $x_0 \in X$*. Let* $\textsc{Adv}$ *be a lazy advice algorithm, and let* $\textsc{Rob}$ *be a lazy version of the work function algorithm. For any* $\epsilon > 0$*, Algorithm 1 (*$\textsc{Dart}$*) is* $(1+\epsilon)$*-consistent and* $\mathcal{O}(\frac{k}{\epsilon})$*-robust.*

This is the *first* result obtaining $(1+\epsilon)$-consistency together with robustness linear in $\frac{1}{\epsilon}$ for k-server; in particular, applying the diameter bound from Lemma 5.2 to the multiplicative weights algorithm of Antoniadis et al. (2020) yields only a bound of $\mathcal{O}(k)$ on both robustness and consistency, which is no better than ignoring advice.

***k*-chasing.** Finally, we consider k-chasing of convex, α-polyhedral functions on $\mathbb{R}$. We assume that $\textsc{Rob}$ is a k-server algorithm that operates on the sequence of minimizers $v_1, \dots, v_T$, e.g., the work function algorithm applied to this sequence, which by Proposition 2.5 is $\mathcal{O}(\frac{k}{\alpha})$-competitive. Moreover, we assume that $\textsc{Adv}$ and $\textsc{Rob}$ are both *lazy*, meaning that they move at most a single server, and they only move a server if it results in strictly lower service cost. Again, this is without loss of generality. A similar diameter bound to that for k-server yields the following result, which is proved in Supplementary Section E.3.

Theorem 5.4. *Let* $\textsc{Adv}$ *be a lazy advice algorithm for* k*-chasing convex,* α*-polyhedral functions on* $\mathbb{R}$*, and let* $\textsc{Rob}$ *be a lazy,* $\mathcal{O}(\frac{k}{\alpha})$*-competitive algorithm for the problem with the property that, at each time* $t \in [T]$*,* $\textsc{Rob}$ *has a server at the minimizer* v_t *of the current cost function. Suppose* $\textsc{Adv}$ *and* $\textsc{Rob}$ *begin with all servers at the same position* $x_0 \in \mathbb{R}$*. Then* $\textsc{Dart}$ *achieves, for any* $\epsilon > 0$*,* $(1+\epsilon)$*-consistency and* $\mathcal{O}(\frac{k}{\alpha\epsilon})$*-robustness.*

6 CONCLUDING REMARKS

We examine the problem of designing learning-augmented algorithms for MTS and its special cases. Our algorithm, DART, achieves $(1+\epsilon)$-consistency and robustness exponential in $\frac{1}{\epsilon}$ for MTS and its special cases, which we show is tight for both MTS and for CFC with a certain weighted ℓ^1 norm. We further show that DART achieves improved performance, matching known results, when the diameter of the problem instance is bounded, and improves upon the best known bounds on robustness and consistency for k-server on any metric space and for k-chasing on the line.

Several interesting avenues remain open for study. Specifically, (i) can subexponential robustness be achieved for CFC and k-chasing with "nicer" norms, e.g., in the Euclidean setting, and (ii) can matching lower bounds be obtained on robustness and consistency for the k-server problem?

Acknowledgements

The authors acknowledge support from an NSF Graduate Research Fellowship (DGE-1745301), NSF grants CNS-2146814, CPS-2136197, CNS-2106403, and NGSDI-2105648, Amazon AWS, and Beyond Limits.

References

Albers, S. and Quedenfeld, J. (2021). Algorithms for Right-Sizing Heterogeneous Data Centers. In *Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '21, pages 48–58, New York, NY, USA. Association for Computing Machinery.

Antoniadis, A., Coester, C., Elias, M., Polak, A., and Simon, B. (2020). Online metric algorithms with untrusted predictions. In *Proceedings of the 37th International Conference on Machine Learning*, pages 345–355. PMLR. ISSN: 2640-3498.

Argue, C. J., Gupta, A., and Guruganesh, G. (2020). Dimension-Free Bounds for Chasing Convex Functions. In *Proceedings of Thirty Third Conference on Learning Theory*, pages 219–241. PMLR. ISSN: 2640-3498.

Bansal, N., Buchbinder, N., Madry, A., and Naor, J. S. (2015a). A Polylogarithmic-Competitive Algorithm for the k-Server Problem. *Journal of the ACM*, 62(5):40:1–40:49.

Bansal, N., Gupta, A., Krishnaswamy, R., Pruhs, K., Schewior, K., and Stein, C. (2015b). A 2-Competitive Algorithm For Online Convex Optimization With Switching Costs. In Garg, N., Jansen, K., Rao, A., and Rolim, J. D. P., editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 96–109, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISSN: 1868-8969.

Blum, A. and Burch, C. (1997). On-line learning and the metrical task system problem. In *Proceedings of the tenth annual conference on Computational learning theory - COLT '97*, pages 45–53, Nashville, Tennessee, United States. ACM Press.

Borodin, A. and El-Yaniv, R. (2005). *Online Computation and Competitive Analysis*. Cambridge University Press.

Borodin, A., Linial, N., and Saks, M. E. (1992). An optimal on-line algorithm for metrical task system. *Journal of the ACM*, 39(4):745–763.

Bubeck, S., Coester, C., and Rabani, Y. (2022). The Randomized k-Server Conjecture is False! arXiv:2211.05753.

Bubeck, S., Cohen, M. B., Lee, J. R., and Lee, Y. T. (2021a). Metrical Task Systems on Trees via Mirror Descent and Unfair Gluing. *SIAM Journal on Computing*, 50(3):909–923.

Bubeck, S., Cohen, M. B., Lee, Y. T., Lee, J. R., and Madry, A. (2018). k-server via multiscale entropic regularization. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 3–16, New York, NY, USA. Association for Computing Machinery.

Bubeck, S., Klartag, B., Lee, Y. T., Li, Y., and Sellke, M. (2019). Chasing Nested Convex Bodies Nearly Optimally. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 1496–1508. Society for Industrial and Applied Mathematics.

Bubeck, S., Rabani, Y., and Sellke, M. (2021b). Online Multiserver Convex Chasing and Optimization. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 2093–2104. Society for Industrial and Applied Mathematics.

Buchbinder, N., Chen, S., Naor, J. S., and Shamir, O. (2012). Unified Algorithms for Online Learning and Competitive Analysis. In *Proceedings of the 25th Annual Conference on Learning Theory*, pages 5.1–5.18. JMLR Workshop and Conference Proceedings. ISSN: 1938-7228.

Chen, N., Goel, G., and Wierman, A. (2018). Smoothed Online Convex Optimization in High Dimensions via Online Balanced Descent. In *Proceedings of the 31st Conference On Learning Theory*, pages 1574–1594. PMLR. ISSN: 2640-3498.

Cho, E., Myers, S. A., and Leskovec, J. (2011). Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090, San Diego California USA. ACM.

Christianson, N., Handina, T., and Wierman, A. (2022). Chasing Convex Bodies and Functions with Black-Box

Advice. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pages 867–908. PMLR. ISSN: 2640-3498.

Coester, C. and Lee, J. R. (2019). Pure entropic regularization for metrical task systems. In *Proceedings of the Thirty-Second Conference on Learning Theory*, pages 835–848. PMLR. ISSN: 2640-3498.

Daniely, A. and Mansour, Y. (2019). Competitive ratio vs regret minimization: achieving the best of both worlds. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, pages 333–368. PMLR. ISSN: 2640-3498.

Dehghani, S., Ehsani, S., Hajiaghayi, M., Liaghat, V., and Seddighin, S. (2017). Stochastic k-Server: How Should Uber Work? In Chatzigiannakis, I., Indyk, P., Kuhn, F., and Muscholl, A., editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 126:1–126:14, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISSN: 1868-8969.

Deng, S., Li, J., and Rabani, Y. (2022). Approximation algorithms for clustering with dynamic points. *Journal of Computer and System Sciences*, 130:43–70.

Friedman, J. and Linial, N. (1993). On convex body chasing. *Discrete & Computational Geometry*, 9(3):293–321.

Goel, G. and Wierman, A. (2019). An Online Algorithm for Smoothed Regression and LQR Control. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 2504–2513. PMLR. ISSN: 2640-3498.

Koutsoupias, E. (2009). The k-server problem. *Computer Science Review*, 3(2):105–118.

Koutsoupias, E. and Papadimitriou, C. H. (1995). On the k-server conjecture. *Journal of the ACM*, 42(5):971–983.

Li, Y., Qu, G., and Li, N. (2021). Online Optimization With Predictions and Switching Costs: Fast Algorithms and the Fundamental Limit. *IEEE Transactions on Automatic Control*, 66(10):4761–4768. Conference Name: IEEE Transactions on Automatic Control.

Lin, M., Liu, Z., Wierman, A., and Andrew, L. L. H. (2012). Online algorithms for geographical load balancing. In *2012 International Green Computing Conference (IGCC)*, pages 1–10.

Lin, M., Wierman, A., Andrew, L. L. H., and Thereska, E. (2013). Dynamic Right-Sizing for Power-Proportional Data Centers. *IEEE/ACM Transactions on Networking*, 21(5):1378–1391. Conference Name: IEEE/ACM Transactions on Networking.

Lin, Y. (2022). Personal correspondence.

Lin, Y., Goel, G., and Wierman, A. (2020). Online Optimization with Predictions and Non-convex Losses. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(1):18:1–18:32.

Lindermayr, A., Megow, N., and Simon, B. (2022). Double Coverage with Machine-Learned Advice. In Braverman, M., editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 99:1–99:18, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISSN: 1868-8969.

Lykouris, T. and Vassilvtiskii, S. (2018). Competitive Caching with Machine Learned Advice. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3296–3305. PMLR. ISSN: 2640-3498.

Mahdian, M., Nazerzadeh, H., and Saberi, A. (2012). Online Optimization with Uncertain Information. *ACM Transactions on Algorithms*, 8(1):1–29.

Manasse, M., McGeoch, L., and Sleator, D. (1988). Competitive algorithms for on-line problems. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 322–333, New York, NY, USA. Association for Computing Machinery.

Mitzenmacher, M. and Vassilvitskii, S. (2021). Algorithms with Predictions. In Roughgarden, T., editor, *Beyond the Worst-Case Analysis of Algorithms*, pages 646–662. Cambridge University Press, Cambridge.

Purohit, M., Svitkina, Z., and Kumar, R. (2018). Improving Online Algorithms via ML Predictions. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Ramesh, S. S., Sessa, P. G., Krause, A., and Bogunovic, I. (2022). Movement Penalized Bayesian Optimization with Application to Wind Energy Systems. In *ICML 2022 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, page 23.

Rutten, D., Christianson, N., Mukherjee, D., and Wierman, A. (2022). Online Optimization with Untrusted Predictions. *arXiv:2202.03519 [cs]*. arXiv: 2202.03519.

Santambrogio, F. (2015). *Optimal Transport for Applied Mathematicians*, volume 87 of *Progress in Nonlinear Differential Equations and Their Applications*. Springer International Publishing, Cham.

Sellke, M. (2020). Chasing convex bodies optimally. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '20, pages 1509–1518, USA. Society for Industrial and Applied Mathematics.

Zhang, L., Jiang, W., Lu, S., and Yang, T. (2021). Revisiting Smoothed Online Learning. In *Proceedings of NeurIPS '21*, volume 34, pages 13599–13612.

Optimal robustness-consistency tradeoffs for learning-augmented metrical task systems: Supplementary Material

A Experimental results

In this section, we examine the application of DART to several real-world datasets for the *online caching problem*, a special case of MTS that may be thought of as k-server on a uniform metric space. We specifically use the same datasets, problem instances, and advice predictors used in Antoniadis et al. (2020) in order to compare the performance of DART against their algorithms for learning-augmented MTS.

Datasets. We compare the performance of DART against both the deterministic and randomized algorithms of Antoniadis et al. (2020) on the same two datasets used in their work.

- The BK dataset originates from a now obsolete social network, BrightKite (Cho et al., 2011). We consider the caching problem with instances generated by sequences of user check-in locations, filtered as in Antoniadis et al. (2020) to include the first 100 instances with the longest nontrivial check-in sequences – those which require at least 50 evictions under the optimal policy. We select a cache size of $k = 10$.
- The Citi dataset originates from a bike-sharing program in New York City (`https://citibikenyc.com/system-data`). We consider caching problem instances where each request corresponds to a trip's starting point in the dataset. We consider 12 sequences representing each month of 2017, each comprised of the first 25,000 bike trips of that month. We use a cache size of $k = 100$.

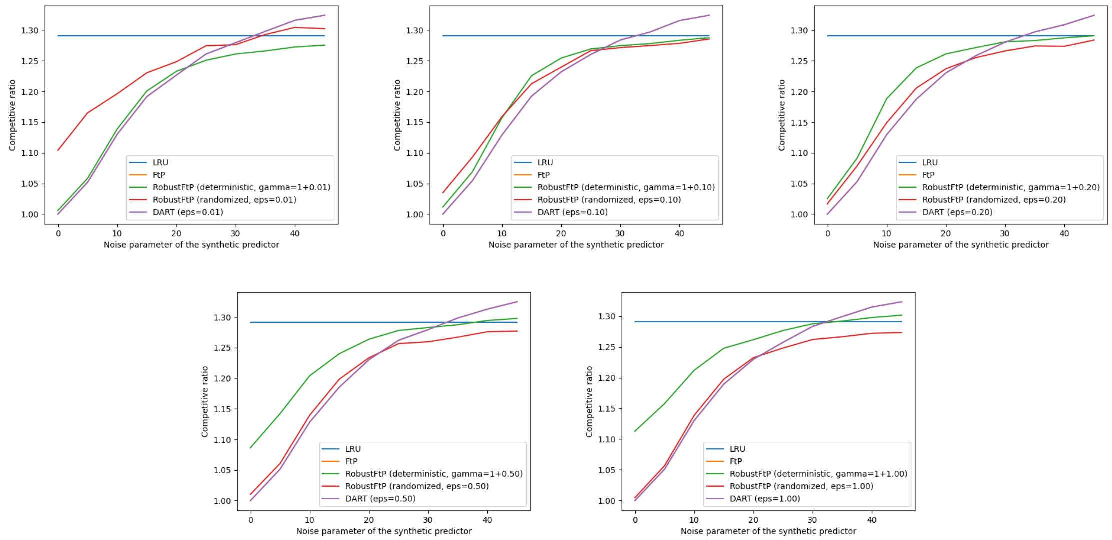

Figure 1: Algorithm performance on the BK dataset with various choices of ϵ.

Advice. We use the same synthetic predictions and advice setup as in Antoniadis et al. (2020) for our experiments. Specifically, for each request, we compute its next arrival time $a(t)$; if it does not appear again in the future, we set $a(t) = T + 1$. To simulate the noise inherent in an ML prediction algorithm, we assume access only to a noisy predictor $p(t) = a(t)+\epsilon$, where $\epsilon \overset{\text{iid}}{\sim} \text{Lognormal}(0, \sigma^2)$ in order to model infrequent but high-impact unreliability of an ML predictor. These noisy predictions of elements' future arrival time are then transformed into an "advice" decision through simulation of an algorithm that uses these predictions to evict the element with the latest predicted arrival time, as well as subsequent filtering with the Follow the Prediction (FtP) algorithm of Antoniadis et al. (2020). This constitutes our advice ADV.

Algorithms. Following Antoniadis et al. (2020), for the "robust" algorithm ROB, we employ Least Recently Used (LRU), a widely used heuristic algorithm for caching that, when faced with a new request, replaces the page that was least recently used by the new page request. We implement the DART algorithm, with ADV the synthetic prediction-based algorithm (which we refer to as FtP) and ROB the LRU algorithm, and compare its performance against both the deterministic and randomized RobustFtP algorithms proposed by Antoniadis et al. (2020) with the same ADV and ROB. We examine the performance of these three algorithms across several choices of ϵ ranging from 0.01 to 1.[2]

Results. We calculate the competitive ratio for each algorithm (including LRU and the advice, denoted FtP) on both datasets by dividing the total number of evicted pages across all instances by the offline optimal number, averaging over 10 independent experiments to account for the randomization of DART and the randomized RobustFtP algorithm. We display the results in Figures 1 and 2. In particular, we observe that on the BK dataset, across all values of ϵ, DART outperforms both the deterministic and randomized RobustFtP algorithms when the magnitude of the noise on the synthetic predictions is small-to-moderate, with this trend reversing as the noise grows large. This behavior is not surprising, since DART uses an *a priori* fixed "trust" parameter ϵ, whereas the randomized RobustFtP algorithm of Antoniadis et al. (2020) essentially runs multiplicative weights on ADV and ROB, and hence is more adaptive in responding to poor advice at the expense of worse exploitation of good advice when the noise is small. The question of adaptively varying DART's parameter ϵ in such a manner to improve performance is an interesting avenue for future work.

On the other hand, we find that on the Citi dataset, DART always slightly outperforms both the deterministic and randomized RobustFtP algorithms. This highlights that, in comparison to the algorithms in Antoniadis et al. (2020), DART yields finer control of the tradeoff between consistency and robustness, since it enables obtaining arbitrarily small consistency (of the form $(1 + \epsilon)$) and doesn't have the diameter-dependent additive term of $\mathcal{O}(D/\epsilon)$ that the randomized RobustFtP algorithm of Antoniadis et al. (2020) has on both the consistency and robustness.

[2]For the deterministic algorithm of Antoniadis et al. (2020), which takes a parameter $\gamma > 1$, we run the algorithm with $\gamma = 1 + \epsilon$ for each choice of ϵ in this range.

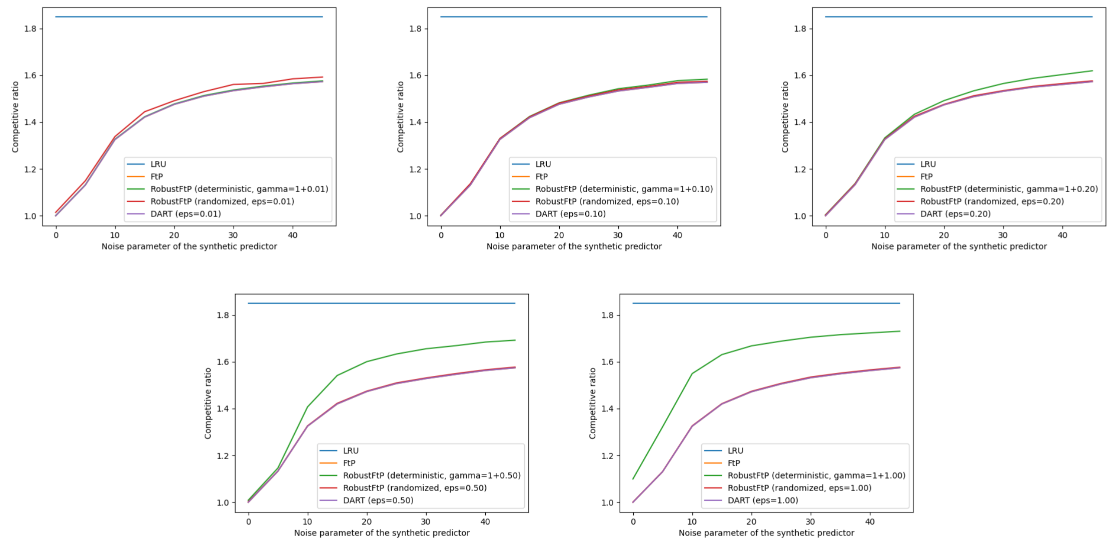

Figure 2: Algorithm performance on the Citi dataset with various choices of ϵ.

B Proof of Proposition 2.5

Let $g_1, \dots, g_T$ be the sequence of α-polyhedral cost functions for an instance of k-chasing, and let $\mathbf{v}_1, \dots, \mathbf{v}_T \in \mathbb{R}^n$ be their minimizers. Let $\textsc{Opt}_s$ be the offline optimal algorithm for the k-server instance on $\mathbb{R}^n$ with requests $\mathbf{v}_1, \dots, \mathbf{v}_T$, and let $\textsc{Opt}_c$ be the offline optimal algorithm for the k-chasing instance on $\mathbb{R}^n$ with function requests $g_1, \dots, g_T$. We denote by $\mathrm{C}^s_{\textsc{Opt}_s}$ the cost of $\textsc{Opt}_s$ as a k-server algorithm (i.e., ignoring the service costs), and by $\mathrm{C}^c_{\textsc{Opt}_s}$ the cost of $\textsc{Opt}_s$ as a k-chasing algorithm (including the hitting costs); we use similar notation for $\textsc{Alg}$. Note that the cost of a k-server algorithm applied to the minimizers of the k-chasing instance will simply be the cost of the k-server algorithm (i.e., the total movement cost incurred by the servers) plus the sum of minimizer costs $\sum_{t=1}^T g_t(\mathbf{v}_t)$, since the minimizer will be occupied by a server at each time. Thus,

$$\mathrm{C}^c_{\textsc{Alg}} = \mathrm{C}^s_{\textsc{Alg}} + \sum_{t=1}^T g_t(\mathbf{v}_t),$$

and the same holds for $\textsc{Opt}_s$.

Let $\mathbf{o}_1, \dots, \mathbf{o}_T \in \binom{\mathbb{R}^n}{k}$ be the sequence of server positions of the algorithm $\textsc{Opt}_c$, and let $i_t \in [k]$ denote the server of $\textsc{Opt}_c$ that realizes the binding service cost at time t, i.e., $i_t := \arg\min_{i \in [k]} g_t(\mathbf{o}_t^{(i)})$. Thus $\mathrm{C}_{\textsc{Opt}_c} = \sum_{t=1}^T g_t(\mathbf{o}_t^{(i_t)}) + d(\mathbf{o}_t, \mathbf{o}_{t-1})$, where d is the minimal matching distance inherited from the norm $\|\cdot\|$. Define the offline algorithm $\textsc{Opt}'_c$ that acts like $\textsc{Opt}_c$, except that at time t it moves the server i_t from $\mathbf{o}_t^{(i_t)}$ to the minimizer $\mathbf{v}_t$, and moves it back to $\mathbf{o}_t^{(i_t)}$ at time $t+1$ before any other server is moved. Since the costs g_t are α-polyhedral,

$$\begin{aligned}
\mathrm{C}_{\textsc{Opt}'_c} &= \sum_{t=1}^T g_t(\mathbf{v}_t) + 2\|\mathbf{o}_t^{(i_t)} - \mathbf{v}_t\| + d(\mathbf{o}_t, \mathbf{o}_{t-1}) \\
&\le \sum_{t=1}^T \max\left\{1, \frac{2}{\alpha}\right\} \left(g_t(\mathbf{v}_t) + \alpha \|\mathbf{o}_t^{(i_t)} - \mathbf{v}_t\| \right) + d(\mathbf{o}_t, \mathbf{o}_{t-1}) \\
&\le \max\left\{1, \frac{2}{\alpha}\right\} \sum_{t=1}^T g_t(\mathbf{o}_t^{(i_t)}) + d(\mathbf{o}_t, \mathbf{o}_{t-1}) \\
&= \max\left\{1, \frac{2}{\alpha}\right\} \mathrm{C}_{\textsc{Opt}_c}. \qquad (1)
\end{aligned}$$

Moreover, since $\textsc{Opt}'_c$ always has a server at $\mathbf{v}_t$ at time t, it is a feasible k-server algorithm for the request sequence $\mathbf{v}_1, \dots, \mathbf{v}_T$, and its cost as a k-server algorithm is simply its total cost as a k-chasing algorithm, minus the the sum of minimizer service costs. Thus we have

$$\begin{aligned} \mathrm{C}^c_{\mathrm{ALG}} &= \mathrm{C}^s_{\mathrm{ALG}} + \sum_{t=1}^T g_t(\mathbf{v}_t) \\ &\leq C \cdot \mathrm{C}^s_{\mathrm{OPT}_s} + \sum_{t=1}^T g_t(\mathbf{v}_t) && (2) \\ &\leq C \cdot \mathrm{C}_{\mathrm{OPT}'_c} && (3) \\ &\leq C \cdot \max\left\{1, \frac{2}{\alpha}\right\} \mathrm{C}_{\mathrm{OPT}_c}, && (4) \end{aligned}$$

where (2) follows by C-competitiveness of ALG for k-server, (3) follows from the fact that OPT_s is optimal and OPT'_c is feasible for the k-server instance $\mathbf{v}_1, \dots, \mathbf{v}_T$, and (4) follows (1). Thus ALG is $C \cdot \max\left\{1, \frac{2}{\alpha}\right\}$-competitive for the k-chasing instance. □

C Proofs for Section 3

C.1 Proof of Theorem 3.1

It suffices to prove the bicompetitive bound in the case that ADV and ROB are deterministic algorithms. That is, we prove the following bound on DART's expected cost:

$$\mathbb{E}[\mathrm{C}_{\mathrm{DART}}] \leq \min\{(1+\epsilon)\mathrm{C}_{\mathrm{ADV}}, 2^{\mathcal{O}(1/\epsilon)}\mathrm{C}_{\mathrm{ROB}}\}, \tag{5}$$

where the expectation is over the randomness of DART. The result in its full generality, i.e., when ADV and ROB can be randomized algorithms, follows by the observation that (5) establishes the same bound on the expected cost of DART *conditioned* on a particular pair of realized trajectories $(a_1, \dots, a_T)$, $(r_1, \dots, r_T)$ of ADV and ROB:

$$\mathbb{E}[\mathrm{C}_{\mathrm{DART}} | a_1, \dots, a_T; r_1, \dots, r_T] \leq \min\{(1+\epsilon)\mathrm{C}_{\mathrm{ADV}}, 2^{\mathcal{O}(1/\epsilon)}\mathrm{C}_{\mathrm{ROB}}\},$$

where the expectation is now over the randomness of DART, ADV, and ROB. With this inequality established, the desired result follows immediately by taking the expectation over the behavior of ADV and ROB on both sides and applying the law of total expectation.

In the following, we thus assume that ADV and ROB are deterministic, with decision trajectories $a_1, \dots, a_T$ and $r_1, \dots, r_T$, respectively. All expectations are over the decisions $x_1, \dots, x_T$ made by DART, which are each distributed marginally according to $x_t \sim p_t$, with consecutive distributions jointly distributed according to the optimal transportation plan γ_t between p_{t-1} and p_t.

We begin by proving competitiveness with respect to ADV, i.e., *consistency* of DART. The argument takes the form of a potential function argument, with potential function $\phi_t = \mathbb{E}[d(x_t, a_t)] = (1-\lambda_t)d(r_t, a_t)$. For an arbitrary time t, there are two cases.

(1) Suppose the algorithm follows the case in line 4; then $\lambda_t = 1$, so $x_t = a_t$. Then

$$\begin{aligned} &\mathbb{E}[f_t(x_t) + d(x_t, x_{t-1}) + \phi_t - \phi_{t-1}] \\ &= f_t(a_t) + \lambda_{t-1}d(a_t, a_{t-1}) + (1-\lambda_{t-1})d(a_t, r_{t-1}) + (1-\lambda_t)d(r_t, a_t) - (1-\lambda_{t-1})d(r_{t-1}, a_{t-1}) \\ &\leq f_t(a_t) + d(a_t, a_{t-1}) && (6) \end{aligned}$$

where (6) follows from the triangle inequality applied to $d(a_t, r_{t-1})$.

(2) Suppose the algorithm follows the case in line 6. First, note that since the coupling between x_{t-1} and x_t is done via the optimal transport plan between p_{t-1} and p_t, we can upper bound $\mathbb{E}[d(x_t, x_{t-1})]$ by the expected movement cost under *any* transport plan between p_{t-1} and p_t. In particular, we can use the transport plan in which we first send a probability mass of $\min\left\{\frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})}, \lambda_{t-1}\right\}$ from a_{t-1} to r_{t-1}, resulting in a mass of λ_t at a_{t-1} and of

$(1-\lambda_t)$ at r_{t-1}, followed by sending the entire mass at a_{t-1} to a_t and the entire mass at r_{t-1} to r_t. Upper bounding $\mathbb{E}[d(x_t, x_{t-1})]$ with this transportation plan, we find:

$$\begin{aligned}\mathbb{E}[d(x_t, x_{t-1})] &\leq (1-\lambda_t)d(r_t, r_{t-1}) + \lambda_t d(a_t, a_{t-1}) \\ &\quad + \min\left\{\frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1}, r_{t-1})}, \lambda_{t-1}\right\} d(a_{t-1}, r_{t-1}) \\ &\leq (1-\lambda_t)d(r_t, r_{t-1}) + \lambda_t d(a_t, a_{t-1}) + \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2}. \end{aligned} \tag{7}$$

Second, note that

$$\begin{aligned}(1-\lambda_t)d(r_t, a_t) &\leq (1-\lambda_t)(d(r_t, r_{t-1}) + d(a_t, a_{t-1})) \\ &\quad + \left(1 - \lambda_{t-1} + \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1}, r_{t-1})}\right) d(a_{t-1}, r_{t-1}) \qquad (8) \\ &\leq (1-\lambda_t)(d(r_t, r_{t-1}) + d(a_t, a_{t-1})) \\ &\quad + (1-\lambda_{t-1})d(a_{t-1}, r_{t-1}) + \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2} \qquad (9)\end{aligned}$$

where (8) follows from the triangle inequality and line 7 of the algorithm. Then, by (7) and (9), and noting that $\lambda_t \leq \lambda_{t-1}$ in this case, we have

$$\begin{aligned}&\mathbb{E}[f_t(x_t) + d(x_t, x_{t-1}) + \phi_t - \phi_{t-1}] \\ &= \lambda_t f_t(a_t) + (1-\lambda_t)f_t(r_t) + \mathbb{E}[d(x_t, x_{t-1})] + (1-\lambda_t)d(r_t, a_t) - (1-\lambda_{t-1})d(r_{t-1}, a_{t-1}) \\ &\leq \left(1 + \frac{\epsilon}{2}\right)(f_t(a_t) + d(a_t, a_{t-1})) + 2(f_t(r_t) + d(r_t, r_{t-1}))\end{aligned}$$

Summing cases 1 and 2 over time and using the fact that case 2 only occurs in timesteps t where $\mathrm{C}_{\mathrm{ROB}}(1,t) < \frac{\epsilon}{4}\mathrm{C}_{\mathrm{ADV}}(1,t)$ we obtain

$$\mathbb{E}[\mathrm{C}_{\mathrm{DART}}] \leq (1+\epsilon)\mathrm{C}_{\mathrm{ADV}}.$$

We now turn to proving the competitive bound with respect to ROB, i.e., *robustness*. Let $\tau \in \{0, \dots, T\}$ be the last time index that $\mathrm{C}_{\mathrm{ROB}}(1,\tau) \geq \frac{\epsilon}{4} \cdot \mathrm{C}_{\mathrm{ADV}}(1,\tau)$. Clearly if $\tau = 0$, then $\lambda_t = 0$ for all $t \in [T]$, so DART follows ROB exactly and we are finished. Thus we restrict to the case that $\tau \geq 1$, i.e., $\lambda_t > 0$ for *some* time $t \in [T]$. By the consistency result just presented, we have

$$\begin{aligned}\mathbb{E}[\mathrm{C}_{\mathrm{DART}}] &= \mathbb{E}[\mathrm{C}_{\mathrm{DART}}(1,\tau)] + \mathbb{E}[\mathrm{C}_{\mathrm{DART}}(\tau+1, T)] \\ &\leq (1+\epsilon)\mathrm{C}_{\mathrm{ADV}} + \mathbb{E}[\mathrm{C}_{\mathrm{DART}}(\tau+1, T)] \\ &\leq \frac{4(1+\epsilon)}{\epsilon}\mathrm{C}_{\mathrm{ROB}}(1,\tau) + \mathbb{E}[\mathrm{C}_{\mathrm{DART}}(\tau+1, T)]. \end{aligned} \tag{10}$$

Thus we are faced with the task of bounding $\mathbb{E}[\mathrm{C}_{\mathrm{DART}}(\tau+1, T)]$ in terms of $\mathrm{C}_{\mathrm{ROB}}$. Let $\sigma \geq \tau$ be the last time index at which $\lambda_\sigma > 0$ (it is possible that $\sigma = T$, i.e., that the weights λ_t remain strictly positive from time τ through the end of the instance). Note that, if $\sigma < T - 1$, then at time $\sigma + 1$ the algorithm will move to coinciding with ROB, and from time

$\sigma+2$ onward the algorithm (and its costs) will exactly coincide with ROB. Then the cost of DART during this phase is

$$\begin{aligned}
\mathbb{E}[\mathrm{C}_{\mathrm{DART}}(\tau+1,T)] &= \mathbb{E}[\mathrm{C}_{\mathrm{DART}}(\tau+1,\sigma+1)+\mathrm{C}_{\mathrm{DART}}(\sigma+2,T)] \\
&= \mathbb{E}[\mathrm{C}_{\mathrm{DART}}(\tau+1,\sigma)]+f_{\sigma+1}(r_{\sigma+1})+\mathbb{E}[d(r_{\sigma+1},x_\sigma)]+\mathrm{C}_{\mathrm{ROB}}(\sigma+2,T) \\
&\le \sum_{t=\tau+1}^{\sigma}\mathbb{E}\left[f_t(x_t)+d(x_t,x_{t-1})\right]+\lambda_\sigma d(r_\sigma,a_\sigma)+\mathrm{C}_{\mathrm{ROB}}(\sigma+1,T) \\
&\le \sum_{t=\tau+1}^{\sigma}\lambda_t f_t(a_t)+(1-\lambda_t)f_t(r_t)+(1-\lambda_t)d(r_t,r_{t-1})+\lambda_t d(a_t,a_{t-1}) \\
&\quad+\frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2}+\lambda_\sigma d(r_\sigma,a_\sigma)+\mathrm{C}_{\mathrm{ROB}}(\sigma+1,T) &(11)\\
&\le \sum_{t=\tau+1}^{\sigma}(1-\lambda_t)\mathrm{C}_{\mathrm{ROB}}(t,t)+\left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(t,t)+\lambda_\sigma d(r_\sigma,a_\sigma)+\mathrm{C}_{\mathrm{ROB}}(\sigma+1,T) &(12)\\
&\le \mathrm{C}_{\mathrm{ROB}}(\tau+1,T)+\left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)+\lambda_\sigma d(r_\sigma,a_\sigma) &(13)\\
&\le \mathrm{C}_{\mathrm{ROB}}(\tau+1,T)+\left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)+\mathrm{C}_{\mathrm{ROB}}(1,\sigma)+\mathrm{C}_{\mathrm{ADV}}(1,\sigma) &(14)\\
&\le \mathrm{C}_{\mathrm{ROB}}(\tau+1,T)+\mathrm{C}_{\mathrm{ROB}}(1,\sigma)+\mathrm{C}_{\mathrm{ADV}}(1,\tau)+\left(2+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) \\
&\le \mathrm{C}_{\mathrm{ROB}}(\tau+1,T)+\mathrm{C}_{\mathrm{ROB}}(1,\sigma)+\frac{4}{\epsilon}\mathrm{C}_{\mathrm{ROB}}(1,\tau)+\left(2+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) &(15)
\end{aligned}$$

where (11) follows (7) and the fact that $\lambda_\sigma>0$, so $\lambda_t=\lambda_{t-1}-\frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})}$ exactly for each $t=\tau+1,\dots,\sigma$, (12) follows from $\lambda_t\le\lambda_{t-1}$ for $t=\tau+1,\dots,\sigma$, (14) follows from the triangle inequality applied to $d(r_\sigma,a_\sigma)$, and (15) follows by the assumption that $\mathrm{C}_{\mathrm{ROB}}(1,\tau)\ge\frac{\epsilon}{4}\cdot\mathrm{C}_{\mathrm{ADV}}(1,\tau)$.

All that remains is to upper bound $(2+\frac{\epsilon}{4})\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)$ under the assumption that $\lambda_\sigma>0$. By assumption, $\lambda_\tau=1$, hence

$$\lambda_\sigma=1-\sum_{t=\tau+1}^{\sigma}\frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})}. \tag{16}$$

This begs the question: *given that $\lambda_\sigma>0$, how large can $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)$ be?* To help answer this question, we prove the following lemma.

Lemma C.1. *Let $(y_i)_{i=0}^T$ be a sequence of nonnegative reals with $y_0>0$. Then*

$$\sum_{t=1}^{T}\frac{y_t}{\sum_{i=0}^{t-1}y_i}\ge\log\left(\frac{\sum_{i=0}^{T}y_i}{y_0}\right). \tag{17}$$

This lemma can be seen as a generalization of the classical observation that the Tth harmonic number H_T is lower bounded by $\log(T+1)$; indeed, this result can be recovered from Lemma C.1 by setting $y_i=1$ for all i. The proof of the lemma goes as follows.

Proof. Define a piecewise constant function $y(t):[0,T]\to\mathbb{R}_+$ as follows:

$$y(t)=\begin{cases}y_1 & \text{for } t\in[0,1)\\ y_2 & \text{for } t\in[1,2)\\ \vdots & \\ y_T & \text{for } t\in[T-1,T]\end{cases}$$

and further define a function $Y(t):[0,T]\to\mathbb{R}_+$ as its integral:

$$Y(t)=y_0+\int_0^t y(s)\,\mathrm{d}x.$$

Note that for $t \in [T]$, $Y(t) = \sum_{i=0}^{t} y_i$. Moreover, by the fundamental theorem of calculus, Y'(t) = y(t).

Since $Y(t)$ is increasing, observe that for arbitrary $t \in [T]$,

$$\int_{t-1}^{t} \frac{y(s)}{Y(s)}\,\mathrm{d}s \le \int_{t-1}^{t} \frac{y_t}{Y(t-1)}\,\mathrm{d}s = \frac{y_t}{Y(t-1)} = \frac{y_t}{\sum_{i=0}^{t-1} y_i}.$$

Thus, we may lower bound the sum on the left-hand side of (17) as follows:

$$\begin{aligned}\sum_{t=1}^{T} \frac{y_t}{\sum_{i=0}^{t-1} y_i} &\ge \sum_{t=1}^{T} \int_{t-1}^{t} \frac{y(s)}{Y(s)}\,\mathrm{d}s \\ &= \int_{0}^{T} \frac{y(s)}{Y(s)}\,\mathrm{d}s \\ &= \int_{0}^{T} \frac{Y'(s)}{Y(s)}\,\mathrm{d}s \\ &= [\log(Y(s))]_{s=0}^{T} \\ &= \log(Y(T)) - \log(Y(0)),\end{aligned}$$

establishing the desired bound. □

With the lemma proved, let us return to (16) and the question of how large $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)$ can be given that λ_σ remains strictly positive. By (16), this is equivalent to the question of how large $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)$ can be given that the sum $\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})}$ is strictly less than 1. To answer this question, it suffices to prove a lower bound on the sum in terms of $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)$. If we can show that

$$\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})} \ge g(\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)) \tag{18}$$

for some strictly increasing function $g : \mathbb{R}_+ \to \mathbb{R}_+$, then $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) \ge g^{-1}(1)$ would imply that $\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})} \ge 1$. Thus the assumption that $\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})} < 1$ will in turn imply an upper bound of $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) < g^{-1}(1)$ on the cost, as desired.

Let us thus construct a lower bound in the form of (18). Before moving on, we note two inequalities: first,

$$d(a_\tau, r_\tau) \le \mathrm{C}_{\mathrm{ADV}}(1,\tau) + \mathrm{C}_{\mathrm{ROB}}(1,\tau) \le \left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\mathrm{ROB}}(1,\tau) \tag{19}$$

by the assumption that $\mathrm{C}_{\mathrm{ROB}}(1,\tau) \ge \frac{\epsilon}{4}\cdot \mathrm{C}_{\mathrm{ADV}}(1,\tau)$. Second, for $t \in \{\tau+1,\ldots,\sigma\}$,

$$\begin{aligned}d(a_t, r_t) &\le \mathrm{C}_{\mathrm{ADV}}(1,t) + \mathrm{C}_{\mathrm{ROB}}(1,t) \\ &\le \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(1,t) && (20)\\ &\le \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(1,\tau) + \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,t) \\ &\le \left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\mathrm{ROB}}(1,\tau) + \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,t), && (21)\end{aligned}$$

where (20) and (21) both follow from the assumption that τ is the last time index in which $\mathrm{C}_{\mathrm{ROB}}(1,\tau) \ge \frac{\epsilon}{4}\cdot\mathrm{C}_{\mathrm{ADV}}(1,\tau)$. Applying the bounds (19) and (21), we obtain

$$\begin{aligned}\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)+(1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})} &\ge \frac{\epsilon}{4}\sum_{t=\tau+1}^{\sigma} \frac{\mathrm{C}_{\mathrm{ADV}}(t,t)}{d(a_{t-1},r_{t-1})} \\ &\ge \frac{\epsilon}{4(1+\frac{\epsilon}{4})}\sum_{t=\tau+1}^{\sigma} \frac{(1+\frac{\epsilon}{4})\mathrm{C}_{\mathrm{ADV}}(t,t)}{(1+\frac{4}{\epsilon})\,\mathrm{C}_{\mathrm{ROB}}(1,\tau) + (1+\frac{\epsilon}{4})\mathrm{C}_{\mathrm{ADV}}(\tau+1,t-1)}. && (22)\end{aligned}$$

(Recall that $\mathrm{C}_{\mathrm{ALG}}(t,t')$ is defined to be 0 when $t' < t$). Applying Lemma C.1 to (22) with $y_0 = \left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\mathrm{ROB}}(1,\tau)$ and $y_i = (1+\frac{\epsilon}{4})\mathrm{C}_{\mathrm{ADV}}(\tau+i,\tau+i)$ for $i = 1,\dots,\sigma-\tau$, we obtain

$$\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1},r_{t-1})} \geq \frac{\epsilon}{4(1+\frac{\epsilon}{4})}\log\left(\frac{\left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\mathrm{ROB}}(1,\tau) + (1+\frac{\epsilon}{4})\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)}{\left(1+\frac{4}{\epsilon}\right)\mathrm{C}_{\mathrm{ROB}}(1,\tau)}\right)$$
$$= \frac{\epsilon}{4+\epsilon}\log\left(1+\frac{\epsilon}{4}\frac{\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma)}{\mathrm{C}_{\mathrm{ROB}}(1,\tau)}\right). \tag{23}$$

Thus the lower bound (18) holds with $g:\mathbb{R}_+\to\mathbb{R}_+$ defined as $g(y) = \frac{\epsilon}{4+\epsilon}\log\left(1+\frac{\epsilon}{4}\frac{y}{\mathrm{C}_{\mathrm{ROB}}(1,\tau)}\right)$. Since

$$g^{-1}(1) = \frac{4\mathrm{C}_{\mathrm{ROB}}(1,\tau)}{\epsilon}\left[\exp\left(\frac{4+\epsilon}{\epsilon}\right)-1\right],$$

by the argument following (18), we obtain the upper bound $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) < \frac{4\mathrm{C}_{\mathrm{ROB}}(1,\tau)}{\epsilon}\left[\exp\left(\frac{4+\epsilon}{\epsilon}\right)-1\right]$ on ADV's cost from time $\tau+1$ through σ. Substituting this bound into (15), and that bound subsequently into (10), we conclude that

$$\begin{aligned}
\mathbb{E}[\mathrm{C}_{\mathrm{DART}}] &\leq \frac{4(1+\epsilon)}{\epsilon}\mathrm{C}_{\mathrm{ROB}}(1,\tau) + \mathbb{E}[\mathrm{C}_{\mathrm{DART}}(\tau+1,T)] \\
&\leq \frac{4(1+\epsilon)}{\epsilon}\mathrm{C}_{\mathrm{ROB}}(1,\tau) + \mathrm{C}_{\mathrm{ROB}}(\tau+1,T) + \mathrm{C}_{\mathrm{ROB}}(1,\sigma) + \frac{4}{\epsilon}\mathrm{C}_{\mathrm{ROB}}(1,\tau) + \left(2+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) \\
&\leq \left(5+\frac{8}{\epsilon}\right)\mathrm{C}_{\mathrm{ROB}} + \left(2+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) \\
&\leq \left(5+\frac{8}{\epsilon}\right)\mathrm{C}_{\mathrm{ROB}} + \left(2+\frac{\epsilon}{4}\right)\frac{4\mathrm{C}_{\mathrm{ROB}}(1,\tau)}{\epsilon}\left[\exp\left(\frac{4+\epsilon}{\epsilon}\right)-1\right] \\
&= 2^{\mathcal{O}(1/\epsilon)}\mathrm{C}_{\mathrm{ROB}}.
\end{aligned}$$

This concludes the proof. □

C.2 Proof of Corollary 3.1.1

These results follow immediately from Theorem 3.1, the definition of bicompetitiveness (Definition 2.3), and the observation that an algorithm that is (c,r)-bicompetitive with respect to $(\mathrm{ADV},\mathrm{ROB})$, where ROB is b-competitive, achieves c-consistency with respect to ADV together with rb robustness. Thus (i) follows from the existence of an $\mathcal{O}(\log^2 n)$-competitive algorithm for MTS on any n-point metric space (Bubeck et al., 2021a); (ii) follows from the existence of an n-competitive algorithm for CFC on any n-dimensional normed vector space (Sellke, 2020), as well as the fact that CFC algorithms can be derandomized by taking the expectation; (iii) follows from the fact that the work function algorithm is $(2k-1)$-competitive for k-server (Koutsoupias and Papadimitriou, 1995); and (iv) follows from Proposition 2.5, i.e. the fact that the work function algorithm is $(2k-1)\max\left\{1,\frac{2}{\alpha}\right\}$-competitive for k-chasing α-polyhedral convex functions.

D Proofs for Section 4

D.1 Proof of Theorem 4.1

We proceed under the assumption that $\frac{2}{\epsilon}\in\mathbb{N}$; if this is not the case, then the same result holds up to some small constant factor. We define the metric space (X,d) as follows: $X = \{0\}\cup\{2^i : i = 0,\dots,\frac{2}{\epsilon}\}$, and d is just the usual (Euclidean) metric on $\mathbb{R}$: for $x,y\in X$, $d(x,y) = |x-y|$. All algorithms start at $x_0 = 1$.

The MTS instance realizing the lower bound is constructed as follows: at each time $t = 1,\dots,T := \frac{2}{\epsilon}$, the adversary delivers the service cost function

$$f_t(x) = \infty\cdot\mathbb{1}_{x\notin\{0,2^t\}},$$

forcing any competitive algorithm to assign zero probability mass to any point other than 0 and 2^t. The advice chooses decisions $\mathrm{ADV}_t = 2^t$ at each time, i.e., it deterministically chooses the rightmost point with zero service cost. Let ALG be an arbitrary randomized algorithm for MTS that is $(1+\epsilon)$-consistent with respect to ADV.

Suppose p_t is the probability that $\textsc{Alg}$ assigns to the state $\textsc{Adv}_t = 2^t$ at time t; $1 - p_t$ is thus the probability assigned to the state 0. If $p_t \le p_{t-1}$, then the expected movement cost of $\textsc{Alg}$ at time t is

$$\mathbb{W}_X^1(p_t, p_{t-1}) = p_t 2^{t-1} + (p_{t-1} - p_t)2^{t-1} = p_{t-1}2^{t-1}.$$

On the other hand, if $p_t > p_{t-1}$, then the expected movement cost of $\textsc{Alg}$ at time t is

$$\mathbb{W}_X^1(p_t, p_{t-1}) = p_{t-1}2^{t-1} + (p_t - p_{t-1})2^t \ge p_t 2^{t-1} > p_{t-1}2^{t-1}.$$

Combining the above equality and inequality, the total cost of $\textsc{Alg}$ from time 1 through t is bounded below as

$$\mathbb{E}[\mathrm{C}_{\textsc{Alg}}(1,t)] \ge \sum_{\tau=1}^{t} p_{\tau-1}2^{\tau-1} \tag{24}$$

for any $t \in [T]$, with $p_0 = 1$ by convention.

Since $\textsc{Alg}$ is $(1+\epsilon)$-consistent with respect to $\textsc{Adv}$, it must be the case that, for each $t \in [T]$,

$$\mathbb{E}[\mathrm{C}_{\textsc{Alg}}(1,t)] + (1 - p_t)2^t \le (1+\epsilon)\mathrm{C}_{\textsc{Adv}}(1,t). \tag{25}$$

If this were not the case, then the adversary could simply send $f_{t+1}(x) = \infty \cdot \mathbb{1}_{x \neq 2^t}$ as the final service cost and end the instance, and $\textsc{Alg}$ would violate the assumed consistency. Note that $\mathrm{C}_{\textsc{Adv}}(1,t) = \sum_{\tau=1}^{t} 2^{\tau-1} = 2^t - 1$. By the inequalities (24) and (25), it must hold that

$$\sum_{\tau=1}^{t} p_{\tau-1}2^{\tau-1} + (1-p_t)2^t \le (1+\epsilon)(2^t - 1) \tag{26}$$

for all $t \in [T]$. For $t = 1$, this tells us that $1 + 2(1-p_1) \le 1+\epsilon$, so $p_1 \ge 1 - \frac{\epsilon}{2}$. It is straightforward to see via induction that in general, $p_t \ge 1 - t\frac{\epsilon}{2}$. Thus, from (24), we obtain

$$\begin{aligned}
\mathbb{E}[\mathrm{C}_{\textsc{Alg}}] &\ge \sum_{t=1}^{T} p_{t-1}2^{t-1} \\
&\ge \sum_{t=1}^{T} \left(1 - (t-1)\frac{\epsilon}{2}\right) 2^{t-1} \\
&= \frac{\epsilon}{2} 2^{\frac{2}{\epsilon}+1} - (1+\epsilon)
\end{aligned}$$

where the final equality follows from $T = \frac{2}{\epsilon}$. Thus we have obtained that $\mathbb{E}[\mathrm{C}_{\textsc{Alg}}] = 2^{\Omega(1/\epsilon)}$. Since the offline optimal algorithm for this instance simply moves to 0 and stays there, incurring total cost 1, $\textsc{Alg}$ is thus $2^{\Omega(1/\epsilon)}$-robust. □

D.2 Proof of Theorem 4.2

The proof proceeds via a reduction to the lower bound presented in the previous proof (Supplementary Section D.1). Specifically, we show that the space of probability distributions over the metric space (X, d) from the previous proof endowed with the Wasserstein-1 distance $\mathbb{W}_X^1$ is bijectively isometric to a convex subset K of a vector space endowed with a weighted ℓ^1 norm. This fact, along with a similar correspondence between service costs, will imply that any trajectory of decisions produced by a randomized MTS algorithm on a given problem instance is in one-to-one correspondence with a trajectory of decisions produced by a deterministic CFC algorithm on a corresponding instance, and that moreover, the two trajectories incur identical cost (both movement and service). Note that this correspondence was essentially observed for tree metrics in Bubeck et al. (2021a); our construction is slightly different, so we provide further detail for the sake of completeness.

Let $n = \frac{2}{\epsilon} + 2$, and let $X = \{0\} \cup \{2^i : i = 0, \dots, \frac{2}{\epsilon}\}$ be as in the previous section. Let the simplex $\Delta_n \subset \mathbb{R}^n$ represent the set of probability distributions over X, with ith coordinate corresponding to the probability assigned to the ith state of X (in increasing order, e.g., 0 is the 1st state). We define the convex body

$$K = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \ge 0, x_1 = 1, x_i \ge x_{i+1} \text{ for } i = 1, \dots, n-1\}.$$

Let us define a linear map from Δ_n to K: the map $\mathbf{\Phi} : \mathbb{R}^n \to \mathbb{R}^n$ represented by the upper triangular matrix with all ones on and above the diagonal, and all zeros below the diagonal. In other words,

$$(\mathbf{\Phi}\mathbf{p})_i = \sum_{j \geq i} p_j$$

for each $i \in [n]$. It is straightforward to observe that $\mathbf{\Phi}(\Delta_n) \subseteq K$, by the property that any $\mathbf{p} \in \Delta_n$ satisfies $\mathbf{p} \geq 0$ and $\mathbf{1}^\top \mathbf{p} = 1$. To see that $\mathbf{\Phi}^{-1}(K) \subseteq \Delta_n$, first note that $\mathbf{\Phi}^{-1}$ is just the matrix with 1s occupying its diagonal, and -1s just above the diagonal, i.e.,

$$\mathbf{\Phi}^{-1} = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix}.$$

But then, for $\mathbf{x} \in K$,

$$\mathbf{\Phi}^{-1}\mathbf{x} = \begin{bmatrix} x_1 - x_2 \\ \vdots \\ x_{n-1} - x_n \\ x_n \end{bmatrix}$$

And thus by definition of K, we have $\mathbf{\Phi}^{-1}\mathbf{x} \geq 0$ and $\mathbf{1}^\top \mathbf{\Phi}^{-1}\mathbf{x} = x_1 = 1$. Thus $\mathbf{\Phi}^{-1}(K) \subseteq \Delta_n$, so $\mathbf{\Phi}$ is a bijection between Δ_n and K.

Now, define a vector of weights $\mathbf{w} \in \mathbb{R}^n$ with $w_1 = w_2 = 1$, and $w_i = 2^{i-3}$ for $i = 3, \dots, n$. We define a correspondingly weighted ℓ^1 norm as follows: for $\mathbf{x} \in \mathbb{R}^n$,

$$\|\mathbf{x}\|_{\ell^1(\mathbf{w})} := \sum_{i=1}^n w_i |x_i|.$$

On the other hand, we also consider the Wasserstein-1 distance

$$\mathbb{W}^1_X(\mathbf{p}, \mathbf{p}') = \min_{\gamma \in \Pi(\mathbf{p}, \mathbf{p}')} \mathbb{E}_{(x,x') \sim \gamma}[d(x, x')]$$

between two distributions $\mathbf{p}, \mathbf{p}' \in \Delta_n$ over states of X. Since X is a subset of $\mathbb{R}$ and d is the standard metric on $\mathbb{R}$, the Wasserstein-1 distance can be computed in closed form (Santambrogio, 2015): defining $F_{\mathbf{p}} : \mathbb{R} \to [0,1]$ as the cumulative distribution function of $\mathbf{p}$ over $\mathbb{R}$, we have

$$\mathbb{W}^1_X(\mathbf{p}, \mathbf{p}') = \int_{\mathbb{R}} |F_{\mathbf{p}}(t) - F_{\mathbf{p}'}(t)| \, \mathrm{d}t = |p_1 - p'_1| + \sum_{i=2}^n 2^{i-2} \sum_{j=1}^i |p_j - p'_j|. \tag{27}$$

We now show that $\mathbf{\Phi}$ preserves the Wasserstein-1 distance: for any $\mathbf{p}, \mathbf{p}' \in \Delta_n$, we have

$$\|\mathbf{\Phi}\mathbf{p} - \mathbf{\Phi}\mathbf{p}'\|_{\ell^1(\mathbf{w})} = \sum_{i=1}^n w_i \left| \sum_{j=i}^n p_j - p'_j \right|. \tag{28}$$

Applying the equalities $\mathbf{1}^\top \mathbf{p} = \mathbf{1}^\top \mathbf{p}' = 1$ and $\sum_{j=i}^n p_j = 1 - \sum_{j=1}^{i-1} p_j$ (and similarly for $\mathbf{p}'$), equality of (27) and (28) follows immediately. Thus $\mathbf{\Phi}$ is a bijective isometry between $(\Delta_n, \mathbb{W}^1_X)$ and $(K, \|\cdot\|_{\ell^1(w)})$.

We now go about showing that on the MTS instance realizing the lower bound proved in the previous section (Supplementary Section D.1), there is a corresponding CFC instance with the property that any sequence of decisions $\mathbf{p}_1, \dots, \mathbf{p}_T \in \Delta_n$ for the MTS instance maps under $\mathbf{\Phi}$ to a sequence of decisions $\mathbf{x}_1, \dots, \mathbf{x}_T$ for the CFC instance, and that moreover, these sequences have identical costs for their respective instances. Note that this correspondence will hold more generally beyond the particular instance we consider.

Define for each $t \in [T]$ the vector $\mathbf{c}_t \in \overline{\mathbb{R}}^n_+$ whose 1st and $(t+2)$th entry is 0, with all other entries $+\infty$; these are the vector representations of the service cost functions for the lower bound from the previous section. Then let us define a CFC instance with cost functions $f_t : \mathbb{R}^n \to \overline{\mathbb{R}}_+$ defined as

$$f_t(\mathbf{x}) = \mathbf{c}_t^\top \mathbf{\Phi}^{-1}\mathbf{x} + \infty \cdot \mathbb{1}_{\mathbf{x} \notin K}.$$

The costs f_t are certainly convex, since $\mathbf{c}_t^\top \mathbf{\Phi}^{-1}\mathbf{x}$ is linear and K is a convex set. Moreover, because of the indicator term, the only decisions yielding finite cost are those residing in K. Observe that for any $\mathbf{p} \in \Delta_n$, $f_t(\mathbf{\Phi}\mathbf{p}) = \mathbf{c}_t^\top \mathbf{p}$. Thus, it is straightforward to observe by the construction of the cost functions and the fact that $\mathbf{\Phi}$ is a bijective isometry between $(\Delta_n, \mathbb{W}_X^1)$ and $(K, \|\cdot\|_{\ell^1(w)})$ that the CFC instance defined by $f_1, \dots, f_T$ on $\mathbb{R}^n$ is equivalent to the MTS instance defined by $\mathbf{c}_1, \dots, \mathbf{c}_T$ on X, in the sense that sequences of decisions for the latter are in one-to-one correspondence via $\mathbf{\Phi}$ with (finite-cost) sequences of decisions for the former, and this correspondence preserves total cost (both moving and service). Thus any performance bound on algorithms for the MTS instance translates to an identical performance bound on algorithms for the CFC instance, giving the desired result. □

E Proofs for Section 5

E.1 Proof of Theorem 5.1

The proof is identical to that of Theorem 3.1 presented in Supplementary Section C.1, save for the function g realizing the lower bound 18. By assumption, $d(a_t, r_t) \leq D$ for all $t \in [T]$, hence

$$\begin{aligned}\sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t) + (1-\lambda_{t-1})f_t(a_t)}{2d(a_{t-1}, r_{t-1})} &\geq \sum_{t=\tau+1}^{\sigma} \frac{\frac{\epsilon}{2}\mathrm{C}_{\mathrm{ADV}}(t,t)}{2D} \\ &= \frac{\epsilon}{4D}\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma).\end{aligned}$$

Thus, per the argument following (18), $\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) < \frac{4D}{\epsilon}$. Substituting this bound into (13), we obtain

$$\begin{aligned}\mathbb{E}[\mathrm{C}_{\mathrm{DART}}(\tau+1,T)] &\leq \mathrm{C}_{\mathrm{ROB}}(\tau+1,T) + \left(1+\frac{\epsilon}{4}\right)\mathrm{C}_{\mathrm{ADV}}(\tau+1,\sigma) + \lambda_\sigma d(r_\sigma, a_\sigma) \\ &\leq \mathrm{C}_{\mathrm{ROB}}(\tau+1,T) + \left(1+\frac{\epsilon}{4}\right)\frac{4D}{\epsilon} + D\end{aligned}$$

and substituting this bound subsequently into (10), we conclude

$$\begin{aligned}\mathbb{E}[\mathrm{C}_{\mathrm{DART}}] &\leq \frac{4(1+\epsilon)}{\epsilon}\mathrm{C}_{\mathrm{ROB}}(1,\tau) + \mathbb{E}[\mathrm{C}_{\mathrm{DART}}(\tau+1,T)] \\ &\leq \frac{4(1+\epsilon)}{\epsilon}\mathrm{C}_{\mathrm{ROB}}(1,\tau) + \mathrm{C}_{\mathrm{ROB}}(\tau+1,T) + \left(1+\frac{\epsilon}{4}\right)\frac{4D}{\epsilon} + D \\ &\leq \left(4+\frac{4}{\epsilon}\right)\mathrm{C}_{\mathrm{ROB}} + \frac{4D}{\epsilon} + 2D.\end{aligned}$$

Thus the proof. □

E.2 Proof of Lemma 5.2

Recall that all the servers of both ADV and ROB begin at the state $x_0 \in X$ at time 0. Fix any $t \in [T]$. The algorithm ADV has servers at $a_t^{(1)}, \cdots, a_t^{(k)} \in X$ and ROB has servers at $r_t^{(1)}, \cdots, r_t^{(k)}$. Since ADV and ROB are both lazy, each of these $2k$ servers must either be at x_0, or at some previous request s_i for $i \in [t]$. Consider a pair of server positions $a_t^{(j)}$ and $r_t^{(j)}$ for some $j \in [k]$; if $a_t^{(j)} = r_t^{(j)}$, then $d(a_t^{(j)}, r_t^{(j)}) = 0$. On the other hand, if one of the servers is at x_0 and the other is at s_i for some $i \in [t]$, then $d(a_t^{(j)}, r_t^{(j)}) \leq d(x_0, s_i) \leq \mathrm{C}_{\mathrm{OPT}}(1,t)$, since the offline optimal will also have had to move a server from x_0 to meet the request s_i. Finally, if both $a_t^{(j)}$ and $r_t^{(j)}$ are at different requests s_i, s_j for $i \neq j \in [t]$, then

$$d(a_t^{(j)}, r_t^{(j)}) = d(s_i, s_j) \leq \mathrm{C}_{\mathrm{OPT}}(1,t).$$

To see that this holds, note that if OPT served the requests s_i and s_j with different servers, then by the triangle inequality $\mathrm{C}_{\mathrm{OPT}}(1,t) \geq d(x_0, s_i) + d(x_0, s_j) \geq d(s_i, s_j)$ since all the servers began at x_0. On the other hand, if OPT served s_i and s_j with the same server, then that server must have moved from s_i to s_j (or vice versa), hence $d(s_i, s_j) \leq \mathrm{C}_{\mathrm{OPT}}(1,t)$.

Since there are k such pairs of servers $a_t^{(j)}, r_t^{(j)}$, and since d_{mm} is the minimal matching distance, we obtain

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \leq \sum_{j=1}^{k} d(a_t^{(j)}, r_t^{(j)}) \leq k \cdot \mathrm{C}_{\mathrm{OPT}}(1,t).$$

□

E.3 Proof of Theorem 5.4

Before proving the theorem, let us formally define *lazy* algorithms for k-chasing convex, α-polyhedral functions on $\mathbb{R}$.

Definition E.1. *An algorithm* ALG *for k-chasing convex, α-polyhedral functions on $\mathbb{R}$ is **lazy** if, at each time t, the following conditions hold on its decision:*

i. ALG *moves at most a single server at time t, and the only server that it moves (if any) is the one that realizes the service cost.*

ii. ALG *only moves a server in order to obtain a strictly lower service cost.*

That is, if ALG *is a lazy algorithm for k-chasing convex, α-polyhedral functions on $\mathbb{R}$ and $\mathbf{x}_{t-1}, \mathbf{x}_t \in \binom{\mathbb{R}}{k}$ are* ALG*'s decisions at times $t-1$ and t on an instance $g_1, \dots, g_T : \mathbb{R} \to \overline{\mathbb{R}}_+$, then either $\mathbf{x}_{t-1} = \mathbf{x}_t$, or $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ differ by exactly one server, and moreover,*

$$\min_{i \in [k]} g_t(x_t^{(i)}) < \min_{i \in [k]} g_t(x_{t-1}^{(i)}).$$

With this definition formalized, it is straightforward to see by the triangle inequality that, similar to the k-server setting (Borodin and El-Yaniv, 2005, §10.2.3), we can assume without loss of generality that ADV and ROB are lazy algorithms for k-chasing convex functions on $\mathbb{R}$. Next, we prove the following lemma.

Lemma E.2. *Let $g_1, \dots, g_T : \mathbb{R} \to \overline{\mathbb{R}}_+$ be a sequence of α-polyhedral costs for an instance of k-chasing convex, α-polyhedral functions on $\mathbb{R}$ endowed with the usual (Euclidean) metric $d(x, y) = |x - y|$. Let* ADV *and* ROB *be two lazy algorithms for the problem that both start with all servers at the same point $x_0 \in \mathbb{R}$, and let $\mathbf{a}_1, \dots, \mathbf{a}_T \in \binom{\mathbb{R}}{k}$ and $\mathbf{r}_1, \dots, \mathbf{r}_T \in \binom{\mathbb{R}}{k}$ be the sequences of server positions of* ADV *and* ROB*, respectively on the problem instance. Further suppose that* ROB *ends each timestep with a server at the minimizer $v_t = \arg\min_x g_t(x)$, so, $v_t \in \mathbf{r}_t$ at each time t. Then, for any time $t \in [T]$, we have*

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \le \max\{k, k/\alpha\} \cdot \mathrm{C}_{\mathrm{OPT}}(1, t),$$

where d_{mm} is the minimal matching distance inherited from the metric d.

Proof. Suppose without loss of generality that all servers start at $x_0 = 0$. Note that since ADV is lazy (without loss of generality) and costs are α-polyhedral and convex, ADV will never move a server away from the minimizer v_t of the current cost function g_t.

Fix any time $t \in [T]$. Suppose without loss of generality that ADV and ROB have servers indexed in increasing order, i.e., $a_t^{(1)} \le \cdots \le a_t^{(k)}$ and $r_t^{(1)} \le \cdots \le r_t^{(k)}$. Define $\tau = \arg\min_{\tau \in [t]} v_\tau$ and $\sigma = \arg\max_{\sigma \in [t]} v_\sigma$. We break into two cases.

(1.) Suppose $0 \in [v_\tau, v_\sigma]$. Since ADV begins with all servers at 0 and never moves away from a minimizer, all of its servers will lie in the interval $[v_\tau, v_\sigma]$. Similarly, since ROB begins with all servers at 0, is lazy, and always occupies the current minimizer with a server, all of its server positions will also lie in the interval $[v_\tau, v_\sigma]$. As a result, we have

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \le \sum_{i=1}^{k} d(a_t^{(i)}, r_t^{(i)}) \le k \cdot d(v_\tau, v_\sigma), \tag{29}$$

since the minimal matching of $\mathbf{a}_t$ and $\mathbf{r}_t$ will match servers in increasing order, and all servers lie in the interval $[v_\tau, v_\sigma]$.

Now we must simply bound $d(v_\tau, v_\sigma)$ in terms of $\mathrm{C}_{\mathrm{OPT}}(1, t)$. Let let o_τ^* be OPT's closest server to v_τ at time τ, and let o_σ^* be OPT's closest server to v_σ at time σ. Since all servers begin at 0, we can thus lower bound $\mathrm{C}_{\mathrm{OPT}}(1, t)$ by

$$\begin{aligned} \mathrm{C}_{\mathrm{OPT}}(1, t) &\ge g_\tau(o_\tau^*) + d(o_\tau^*, 0) + g_\sigma(o_\sigma^*) + d(o_\sigma^*, 0) \\ &\ge \alpha \cdot d(o_\tau^*, v_\tau) + d(o_\tau^*, 0) + \alpha \cdot d(o_\sigma^*, v_\sigma) + d(o_\sigma^*, 0) \qquad (30) \\ &\ge \min\{1, \alpha\} \cdot d(v_\tau, v_\sigma) \qquad (31) \end{aligned}$$

where (30) follows by α-polyhedrality of the cost functions. Substituting (31) into (29) then gives

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \le \max\left\{k, \frac{k}{\alpha}\right\} \mathrm{C}_{\mathrm{OPT}}(1, t),$$

as desired.

(2.) Suppose 0 is outside of the interval $[v_\tau, v_\sigma]$; we may assume without loss of generality that $0 < v_\tau$. By similar reasoning as in the previous case, all of the servers of ADV and ROB, by laziness, are in the interval $[0, v_\sigma]$. Then we follow a similar argument. Note that

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \leq \sum_{i=1}^{k} d(a_t^{(i)}, r_t^{(i)}) \leq k \cdot d(0, v_\sigma),$$

and, defining o_σ^* as OPT's closest server to v_σ at time σ,

$$\begin{aligned} \mathrm{C}_{\mathrm{OPT}}(1,t) &\geq g_\sigma(o_\sigma^*) + d(o_\sigma^*, 0) \\ &\geq \alpha \cdot d(o_\sigma^*, v_\sigma) + d(o_\sigma^*, 0) \\ &\geq \min\{1, \alpha\} \cdot d(0, v_\sigma). \end{aligned}$$

Thus we obtain

$$d_{\mathrm{mm}}(\mathbf{a}_t, \mathbf{r}_t) \leq \max\left\{k, \frac{k}{\alpha}\right\} \mathrm{C}_{\mathrm{OPT}}(1,t),$$

completing the proof.

□

The result of Theorem 5.4 then follows immediately by substituting the diameter bound from Lemma E.2 into Theorem 5.1 and instantiating ROB with the work function algorithm applied to the minimizer sequence $v_1, \ldots, v_T$, which we know is $\mathcal{O}(\frac{k}{\alpha})$-competitive. □