Movement Penalized Bayesian Optimization with Application to Wind Energy Systems

Shyam Sundhar Ramesh* SHRAMESH@STUDENT.ETHZ.CH
Pier Giuseppe Sessa* SESSAP@ETHZ.CH
Andreas Krause* KRAUSEA@ETHZ.CH
Ilija Bogunovic+ I.BOGUNOVIC@UCL.AC.UK
ETH Zurich*, UCL+

Abstract

Contextual Bayesian optimization (CBO) is a powerful framework for sequential decision-making given side information, with important applications, e.g., in wind energy systems. In this setting, the learner receives context (e.g., weather conditions) at each round, and has to choose an action (e.g., turbine parameters). Standard algorithms assume no cost for switching their decisions at every round. However, in many practical applications, there is a cost associated with such changes, which should be minimized. We introduce the episodic CBO with movement costs problem and, based on the online learning approach for metrical task systems of Coester and Lee [9], propose a novel randomized mirror descent algorithm that makes use of Gaussian Process confidence bounds. We compare its performance with the offline optimal sequence for each episode and provide rigorous regret guarantees. We further demonstrate our approach on the important real-world application of altitude optimization for Airborne Wind Energy Systems. In the presence of substantial movement costs, our algorithm consistently outperforms standard CBO algorithms.

1. Introduction

Bayesian optimization (BO) is a well-established framework for sequential black-box function optimization that relies on Gaussian Process (GP) models [18] to sequentially learn and optimize the unknown objective. In many practical scenarios, however, one wants to additionally use available *contextual* information when making decisions. In this setting, at each round, the learner receives a context from the environment and has to choose an action based upon it. Prior works have developed contextual BO algorithms [16, 7, 15, 17], with various applications, e.g., vaccine design, nuclear fusion, database tuning, etc.

A potential practical issue with such algorithms is that they assume no explicit costs for *switching* between their actions at every round. Frequent action changes can be extremely costly in many applications. This work is motivated by the problem of real-time altitude control of an airborne wind energy (AWE) system.[1] In AWE systems, the wind speed is often only measurable at the system's altitude, and determining the optimal operating altitude of an AWE system as the wind speed varies represents a challenging problem. Another fundamental challenge is that frequent altitude adjustments are costly as it requires additional energy. Consequently, this work is motivated by the following question: *How can we efficiently learn to optimize the AWE system's operating altitude despite varying wind conditions while minimizing the energy cost associated with altitude changes?*

In this work, we formalize the *movement penalized* contextual BO problem. When the switching cost is a *metric* (distance function), we propose a novel algorithm that effectively

1. AWE system is a wind turbine with a rotor supported in the air without a tower that can benefit from the persistence of wind at different high altitudes [12].

combines ideas from BO with the online learning strategies proposed in [9] for solving the so-called *metrical task system* (MTS) problem [5]. Furthermore, our algorithm relies solely on noisy point evaluations (i.e., bandit feedback), allows for arbitrary context sequences, and besides the standard exploration-exploitation trade-off, it also balances the movement costs. As a result, it outperforms the standard movement-cost-agnostic contextual BO algorithms as well as movement-conservative baselines.

2. Problem Statement

Let $f : \mathcal{X} \times \mathcal{E} \to \mathbb{R}_+$ be an *unknown* cost function defined over $\mathcal{X} \times \mathcal{E} \subset \mathbb{R}^p$, where $\mathcal{X}$ is a finite set of actions, i.e., $|\mathcal{X}| = n$, and $\mathcal{E}$ represents convex and compact space of contexts. We denote the *known* metric of $\mathcal{X}$ as $d(\cdot, \cdot)$, and assume that the target cost function f belongs to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ of functions (defined on $\mathcal{X} \times \mathcal{E}$), that corresponds to a known kernel $k : (\mathcal{X} \times \mathcal{E}) \times (\mathcal{X} \times \mathcal{E}) \to \mathbb{R}_+$ with $k((x, e), (x', e')) \leq 1$ for any action-context pair. In particular, we assume that for some known $B > 0$, the target cost f has a bounded RKHS norm, i.e., $f \in \mathcal{F}_k = \{f \in \mathcal{H}_k : \|f\|_k \leq B\}$. Also, we assume that the diameter of $\mathcal{X}$ ($\max_{x,x' \in \mathcal{X}} d(x, x')$) is bounded and denote it by ψ.

We consider an episodic setting, wherein each episode runs over a finite time horizon H. Let the *initial state* of the system in the first episode correspond to action $x_{0,1} \in \mathcal{X}$. At the end of every episode, the system resets to a new given initial action $x_{0,m} \in \mathcal{X}$ where $m \in \{1, 2, \ldots, N_{ep}\}$ denotes the episode index. In each episode m and at every time step $h \in \{1, 2, \ldots, H\}$, the environment reveals the context $e_{h,m} \in \mathcal{E}$ to the learner. We make no assumptions on the context sequence (i.e., it can be arbitrary and different across episodes). The learner then chooses $x_{h,m} \in \mathcal{X}$ and observes the noisy function value:

$$y_{h,m} = f(x_{h,m}, e_{h,m}) + \xi_{h,m}, \tag{1}$$

where $\xi_{h,m} \sim \mathcal{N}(0, \sigma^2)$ with known σ, and independence over time steps. The goal of the learner is to minimize the cost incurred over the rounds in every episode, but at the same time to minimize the distance between its subsequent decisions as measured by $d(x_{h-1,m}, x_{h,m})$.

Let $D_m = \{x_{1,m}, x_{2,m}, \ldots, x_{H,m}\}$ denote the set of actions chosen by the learner over H rounds in episode m. We recall that each action $x_{h,m} \in D_m$ is chosen after observing the corresponding context $e_{h,m}$. The objective is to minimize the cumulative episodic cost for each episode m,

$$\text{cost}_m(D_m) = \underbrace{\sum_{h=1}^{H} f(x_{h,m}, e_{h,m})}_{S_m(D_m)} + \underbrace{\sum_{h=1}^{H} d(x_{h,m}, x_{h-1,m})}_{M_m(D_m)}, \tag{2}$$

where we refer to the two terms in Eq. (2) as *service cost* S_m and *movement cost* M_m.

When f is known, the problem can be seen as a MTS instance as detailed in Section 2.2. Even in such a case, we cannot hope to solve this problem optimally, and nearly-optimal approximate algorithms were recently proposed (see Coester and Lee [9]). Hence, the learner's performance in episode m is measured via (α, β)-approximate regret:

$$r_m^{\alpha,\beta} = \text{cost}_m(D_m) - \alpha \cdot \text{cost}_m(D_m^*) - \beta, \tag{3}$$

where $D_m^* := \arg\min_{D \subset \mathcal{X},\, |D|=H} \text{cost}_m(D)$ is the offline optimal action sequence obtained assuming the knowledge of the true sequence of contexts $\{e_{h,m}\}_{h=1}^H$ *in advance*, and α and β

are approximation constants (independent of N_{ep}). In contrast, in our setting, the learner only gets to see the *current context* when making a decision and has no knowledge about the future ones. After N_{ep} episodes, the total cumulative regret is defined as

$$R_{N_{ep}}^{\alpha,\beta} = \sum_{m=1}^{N_{ep}} r_m^{\alpha,\beta}. \tag{4}$$

We seek an algorithm whose total cumulative regret grows sublinearly in N_{ep}, so that $\lim_{N_{ep}\to\infty} R_{H,N_{ep}}^{\alpha,\beta}/N_{ep} = 0$, for any set of initial states $\{x_{0,m}\}_{m=1}^{N_{ep}} \subset \mathcal{X}$.

2.1 Gaussian Process Model

A Gaussian Process $GP(\mu(\cdot), k(\cdot,\cdot))$ over the input domain $\mathcal{X} \times \mathcal{E}$, is a collection of random variables $(f(x,e))_{x\in\mathcal{X},e\in\mathcal{E}}$ where every finite number of them $(f(x_i,e_i))_{i=1}^n$, $n \in \mathbb{N}$, is jointly Gaussian with mean $\mu(x_i,e_i)$ and covariance $k((x_i,e_i),(x_j,e_j))$ for every $1 \le i,j \le n$.

BO algorithms typically use zero-mean GP priors to model uncertainty in f, i.e., $f \sim GP(0, k(\cdot,\cdot))$, and Gaussian likelihood models for the observed data. As more data points are observed, GP (Bayesian) posterior updates are performed in which noise variables are assumed to be drawn independently across t from $\mathcal{N}(0,\lambda)$. Here, λ is a hyperparameter that might be different from the true noise variance σ^2. More precisely, given the queried points and their noisy observations the posterior is again Gaussian with mean and variance:

$$\mu_t(x,e) = k_t(x,e)^T(K_t + \lambda I_t)^{-1}Y_t \quad \sigma_t^2(x,e) = k((x,e),(x,e)) - k_t(x,e)^T(K_t + \lambda I_t)^{-1}k_t(x,e), \tag{5}$$

where $Y_t := [y_1,\dots,y_t]$ denotes a vector of observations, $K_t = [k((x_s,e_s),(x_{s'},e_{s'})]_{s,s'\le t}$ is the corresponding kernel matrix, and $k_t(x,e) = [k((x_1,e_1),(x,e)),\dots,k((x_t,e_t),(x,e))]^T \in \mathbb{R}^{t\times 1}$.

Maximum Information Gain. In standard BO, the main quantity that characterizes the complexity of optimizing the target cost function is the maximum information gain [19] defined at time t as:

$$\gamma_t = \max_{\{(x_i,e_i)\}_{i=1}^t} I(Y_t; f), \tag{6}$$

where $I(Y_t; f)$ denotes the mutual information between random observations Y_t and GP model f given by $I(Y_t; f) = \frac{1}{2}\log\det(I_t + \lambda^{-1}K_t)$. This quantity is kernel-specific and for compact and convex domains γ_t is sublinear in t for various classes of kernel functions [19] as well as for kernel compositions.

Lemma 1 *([19, 1, 8]) Assume the σ-sub-Gaussian noise model as in Eq. (1), and let f belong to $\mathcal{F}_k$. Then, the following holds with probability at least $1-\delta$ simultaneously over all $t \ge 1$ and $x \in \mathcal{X}$, $e \in \mathcal{E}$: $|\mu_t(x,e) - f(x,e)| \le \beta_t\sigma_t(x,e)$, where $\beta_t = \frac{\sigma}{\lambda^{1/2}}\sqrt{2\ln(1/\delta) + 2\gamma_t} + B$, and μ_t and σ_t are defined in Eq. (5) with $\lambda > 0$.*

Based on Lemma 1, we define the lower confidence bound for every $x \in \mathcal{X}, e \in \mathcal{E}$ as:

$$\mathrm{lcb}_t(x,e) := \mu_t(x,e) - \beta_t\sigma_t(x,e). \tag{7}$$

We use $\mathrm{lcb}_m(x,e)$ when it is computed based on data collected before episode m.

2.2 Relation to Metrical Task Systems (MTS)

When f is known, our optimization objective in Eq. (2) can be seen as a particular type of MTS problem, where $f(\cdot, e_{h,m})$ is the MTS service cost that changes for every h and m. Compared to a standard MTS, our problem formulation is more challenging since the learner can only learn about f from previously observed data. The approach proposed in this paper builds on the algorithm by Coester and Lee [9] for standard MTS problems. However, to

Algorithm 1 GP-MD

Require: Action space $\mathcal{X}$, kernel function $k(\cdot,\cdot)$, metric $d(\cdot,\cdot)$
Run FRT($\mathcal{X}, d(\cdot,\cdot)$) and obtain τ-HST $\mathcal{T}=(V,E)$ with leaves $\mathcal{L}=\mathcal{X}$
for $m=1,\dots,N_{ep}$ **do**
  Receive $x_{0,m}$ and initialize $z_{0,m}$, conditional prob. $q_0=\Delta^{-1}(z_{0,m})$ as in Eq. (11)
  for $h=1,\dots,H$ **do**
    Observe context $e_{h,m}$ and initialize costs: $\mathrm{lcb}_m(v,e_{h,m})=0$, $\forall v\in V\backslash\mathcal{L}$
    for $u\in\mathcal{OD}(V\backslash\mathcal{L})$ **do**
      Update vertex prob. $q_h^{(u)}$ from $q_{h-1}^{(u)}$ and $\mathrm{lcb}_m(\cdot,e_{h,m})$ via Mirror Descent (Eq. (10))
      Update cost for vertex u:

$$\mathrm{lcb}_m(u,e_{h,m})=\langle q_h^{(u)},\mathrm{lcb}_m(\cdot,e_{h,m})\rangle=\sum_{\nu\in\mathcal{C}(u)} q_{h,\nu}\cdot\mathrm{lcb}_m(\nu,e_{h,m})$$

    end for
    Compute prob. vector $z_{h,m}=\Delta(q_h)$ (Eq. (11)) and leaves' prob. $l(z_{h,m})$ (Eq. (8))
    Estimate optimal coupling $\zeta_{h-1,h,m}$ between $l(z_{h-1,m})$ and $l(z_{h,m})$ as in Eq. (9)
    Sample action $x_{h,m}\sim\zeta_{h-1,h,m}(\cdot|x_{h-1,m})$ and observe $y_{h,m}=f(x_{h,m},e_{h,m})+\xi_{h,m}$
  end for
  Update $\mu_{m+1}(\cdot,\cdot)$ and $\sigma_{m+1}(\cdot,\cdot)$ as per Eq. (5)
end for

cope with the aforementioned challenge, our approach exploits the regularity assumptions regarding f and utilizes the constructed lower confidence bounds Eq. (7) to *hallucinate* information about the unavailable service cost at each round. Before presenting our overall approach, we describe a preliminary step proposed by [9], which consists of representing our metric space $(\mathcal{X},d)$ by a *Hierarchically Separated Tree* (HST) metric space.

HST metric space. Consider a tree $\mathcal{T}=(V,E)$ with root r, leaves $\mathcal{L}\subset V$ and non-negative weights w_v, for each $v\in V$, which are non-increasing along root-leaf paths. Let $d_{\mathcal{T}}(l,l')$ denote a distance metric between any two leaves $l,l'\in\mathcal{L}$ given as the sum of the encountered weights on the path from l to l'. $(\mathcal{L},d_{\mathcal{T}})$ is a HST metric space, and τ-HST metric space if the weights are exponentially decreasing, i.e., $w_u\le w_v/\tau$, with v being the parent of u. Similarly to [9], we use the algorithm from [13] (FRT) to approximate the given metric space $(\mathcal{X},d)$ by a τ-HST one with leaves $\mathcal{L}$ corresponding to actions in $\mathcal{X}$.

3. The GP-MD Algorithm

In this section, we introduce GP-MD, a novel algorithm for the contextual BO problem with movement costs defined in Section 2. At each episode m and round h, the state of GP-MD can be summarized by a vector of probabilities $z_{h,m}\in K_{\mathcal{T}}$ over the vertices of $\mathcal{T}$, where $K_{\mathcal{T}}:=\left\{z\in\mathbb{R}_+^{|V|}: z_r=1, z_u=\sum_{\nu\in\mathcal{C}(u)} z_\nu \quad \forall u\in V\backslash\mathcal{L}\right\}$, and $\mathcal{C}(u)$ denotes the children of u. Each entry z_ν represents the probability that the selected action $x_{h,m}$ belongs to the leaves of the subtree rooted at ν, i.e., $z_\nu=\mathbb{P}(x_{h,m}\in\mathcal{L}(\nu))$. Moreover, given any $z\in K_{\mathcal{T}}$,

$$l(z):=[z_l,\ l\in\mathcal{L}]\in[0,1]^n, \tag{8}$$

defines a probability distribution over the leaves $\mathcal{L}$, and hence the actions $\mathcal{X}$. As in [9], given probability vectors $z_{h,m}$ and $z_{h-1,m}$, GP-MD computes the *minimal distance* distribution

$$\zeta_{h-1,h,m} = \underset{\zeta \in \Pi(l(z_{h-1,m}), l(z_{h,m}))}{\arg\inf} \mathbb{E}_{\zeta}[d_{\mathcal{T}}(U_{h-1,m}, U_{h,m})], \tag{9}$$

where $U_{h-1,m}$ and U_h are random variables having marginals $l(z_{h-1,m})$ and $l(z_{h,m})$ respectively. Finally, action $x_{h,m}$ is sampled from the conditional minimal distance distribution $x_{h,m} \sim \zeta_{h-1,h,m}(\cdot|x_{h-1,m})$ (Line 13 in Algorithm 1). At the end of each episode m, the newly observed data are then used to update cost function's posterior mean and variance.

Finally, to compute $z_{h,m}$ (Lines 8–12 in Algorithm 1) we follow the recursive Mirror Descent (MD) procedure proposed by [9], with the important difference that we are dealing with an *unknown* context-dependent cost function. Hence, we make use of the GP model and Eq. (7). To obtain probabilities $z_{h,m}$, we consider *conditional* probability vectors $q \in Q_{\mathcal{T}}$, where $Q_{\mathcal{T}}$ is the set of valid conditional probabilities $Q_{\mathcal{T}} := \Big\{ q \in \mathcal{R}_+^{|V \setminus r|} : \sum_{\nu \in \mathcal{C}(u)} q_\nu = 1 \quad \forall u \in V \setminus \mathcal{L} \Big\}$. For each vertex ν with parent u, q_ν represents the conditional probability $\mathbb{P}(x_{h,m} \in \mathcal{L}(\nu) | x_{h,m} \in \mathcal{L}(u))$. Moreover, given $q_h \in Q_{\mathcal{T}}$ we define the vector $q_h^{(u)} := [q_{h,\nu}, \nu \in \mathcal{C}(u)]$ as the conditional distribution over children of u, and let $Q_{\mathcal{T}}^{(u)}$ be the set of all valid distributions $q_h^{(u)}$. In each episode m, q_h for round h is obtained recursively, from leaves to root, as a function of q_{h-1}, the observed context $e_{h,m}$, and the current estimate about the cost associated to each particular vertex. More precisely, let $\mathcal{OD}(V \setminus \mathcal{L})$ be a topological ordering of the internal vertices $V \setminus \mathcal{L}$ so that every child in $\mathcal{T}$ occurs before its parent. Then, for each $u \in \mathcal{OD}(V \setminus \mathcal{L})$ conditional probabilities $q_h^{(u)}$ are obtained via the Mirror Descent update:

$$q_h^{(u)} = \underset{p \in Q_{\mathcal{T}}^{(u)}}{\arg\min} \Big\{ D^{(u)}(p \| q_{h-1}^{(u)}) + \langle p, \mathrm{lcb}_m^{(u)}(\cdot, e_{h,m}) \rangle \Big\}. \tag{10}$$

Function $D^{(u)}$ is the Bregman divergence with respect to a suitable potential function, while $\mathrm{lcb}_m^{(u)}(\cdot, e_{h,m}) := [\mathrm{lcb}_m(\nu, e_{h,m}),\ \forall \nu \in \mathcal{C}(u)]$ is a lower confidence bound estimate of the costs corresponding to children of vertex u. For $v \in \mathcal{L}$, $\mathrm{lcb}_m(\nu, e_{h,m})$ are obtained by the GP-regression techniques outlined in Section 2.1, while for internal vertices these are computed recursively from their children nodes as: $\mathrm{lcb}_m(u, e_{h,m}) := \sum_{\nu \in \mathcal{C}(u)} q_{h,\nu} \mathrm{lcb}_m(\nu, e_{h,m})$.

Once $q_h \in Q_{\mathcal{T}}$ is updated, we can obtain $z_{h,m}$ via the mapping $\Delta : Q_{\mathcal{T}} \to K_{\mathcal{T}}$ such that:

$$z = \Delta(q) \ \Rightarrow \ z_\nu = z_u q_\nu \quad \forall u \in V \setminus \mathcal{L},\ \nu \in \mathcal{C}(u). \tag{11}$$

Theorem 1 (informal) *The regret of* GP-MD *over N_{ep} episodes is bounded w.h.p. by*

$$R_{N_{ep}}^{\alpha,\beta} = \mathcal{O}\Big(\beta_{N_{ep}} \sqrt{N_{ep} \gamma_{HN_{ep}}}\Big),$$

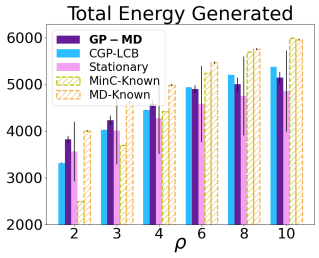
with $\alpha = \mathcal{O}\big((\log n)^2\big)$ and $\beta = \mathcal{O}(1)$.

We observe that the obtained regret bound is sublinear in N_{ep} and hence $\lim_{N_{ep} \to \infty} R_{N_{ep}}^{\alpha,\beta} / N_{ep} = 0$ and is independent of n. These imply that GP-MD approaches α-competitive ratio performance of the MTS algorithm by [9], while learning about the service cost from noisy point evaluations (i.e., bandit feedback) only. Finally, we note that H is treated as constant.

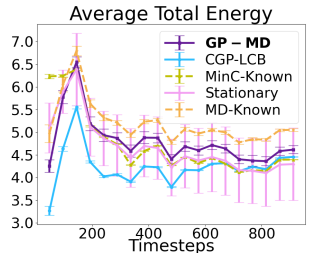
4. Experiments

This section provides numerical results on synthetic and real-world data. We compare the performance of our GP-MD algorithm with the following baselines:

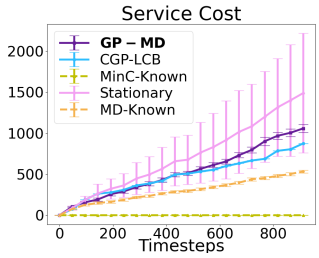
- STATIONARY selects the stationary strategy $x_h = x_0$ for all h,
- CGP-LCB [16] neglects the movement cost and sets $x_h = \arg\min_x \mathrm{lcb}_h(x, e_h)$ for all h,



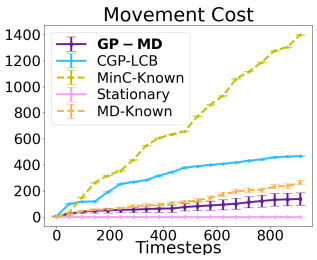
(a) Lat. $= 53$, Long. $= -10$



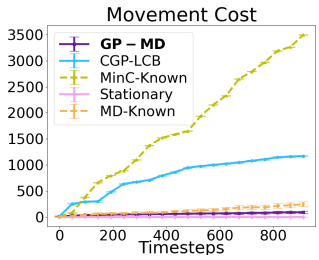
(b) Total Energy ($\rho = 4$)



(c) Service Cost ($\rho = 4$)



(d) Movement Cost ($\rho = 4$)



(e) Movement Cost ($\rho = 1$)

Figure 1: AWE altitude optimization task; Fig. 1a: GP-MD outperforms previously used CGP-LCB (that optimizes for service costs only) for a range of ρ values that favor the service against the movement cost. Fig. 1b: The average total generated energy. Figs. 1c and 1d: The service and movement costs for $\rho = 4$. Fig. 1e: The movement costs for $\rho = 1$. The movement energy loss is slightly lower for GP-MD as compared to $\rho = 4$ due to higher importance towards movement cost reduction.

- MinC-Known assumes $f(\cdot)$ is known and chooses $x_h = \arg\min_x f(x, e_h)$, and
- MD-Known assumes $f(\cdot)$ is known and runs mirror descent from [9] on $f(\cdot, e_h)$.

MD-Known and MinC-Known unrealistically assume that $f(\cdot)$ is known and can be seen as upper-bound for the achievable performance of GP-MD and CGP-LCB, respectively.

Altitude Optimization in AWE Systems: In AWE systems, the turbine's operating altitude can be changed depending on the wind pattern. We follow the setup of Baheri et al. [2] that applied CGP-LCB [16] which ignores movement-costs, to learn this control task. In this section, we use a dataset from [4] which contains wind-speed information over various locations in Europe. Our goal is to maximize the generated energy, while taking into account the energy loss due to altitude change. We consider 25 different altitudes (ranging from 10m to 1600m) as the action space and the context space to be hours in the day (i.e., 24 values). We define $f(x,t) = \max_{x'}(E_S(x',t)) - E_S(x,t)$ where $E_S(x,t)$ denotes the energy generated based on the windspeed at altitude x and time t. We run the algorithms for different ρ for 960 timesteps, where ρ is used to multiply E_S. Based on this we plot the total energy generated w.r.t. varying ρ in Fig. 1a. Our algorithm outperforms CGP-LCB for a range of ρ values. As ρ keeps increasing, we observe that MinC-Known closes the performance gap to MD-Known, and the same is happening with CGP-LCB w.r.t. GP-MD. In Fig. 1b, we focus on a particular $\rho = 4$, and notice that GP-MD performs better than CGP-LCB and Stationary algorithm. In Fig. 1c, we plot the service cost and observe that both learning algorithms GP-MD and CGP-LCB have lower service cost than the Stationary baseline. We also note that due to the implicit service cost definition, the MinC-Known baseline achieves zero service cost. From Figs. 1d and 1e, it is evident that $\rho = 1$ results in slightly lower GP-MD movement energy loss due to the tradeoff shifting towards the movement cost.

References

[1] Abbasi-Yadkori. Online learning for linearly parametrized control problems. 2013.

[2] Ali Baheri, Shamir Bin-Karim, Alireza Bafandeh, and Christopher Vermillion. Real-time control using Bayesian optimization: A case study in airborne wind energy systems. *Control Engineering Practice*, 2017.

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

[4] Philip Bechtle, Mark Schelbergen, Roland Schmehl, Udo Zillmann, and Simon Watson. Airborne wind energy resource analysis. *Renewable energy*, 2019.

[5] Allan Borodin, Nathan Linial, and Michael E Saks. An optimal on-line algorithm for metrical task system. *Journal of the ACM (JACM)*, 1992.

[6] Sébastien Bubeck, Michael B Cohen, James R Lee, and Yin Tat Lee. Metrical task systems on trees via mirror descent and unfair gluing. *SIAM Journal on Computing*, 2021.

[7] Ian Char, Youngseog Chung, Willie Neiswanger, Kirthevasan Kandasamy, Andrew O Nelson, Mark Boyer, Egemen Kolemen, and Jeff Schneider. Offline contextual Bayesian optimization. *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[8] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. *International Conference on Machine Learning (ICML)*, 2017.

[9] Christian Coester and James R Lee. Pure entropic regularization for metrical task systems. *Conference on Learning Theory (COLT)*, 2019.

[10] Christian Coester and James R Lee. Pure entropic regularization for metrical task systems. *arXiv preprint arXiv:1906.04270*, 2019.

[11] Thomas Desautels, Andreas Krause, and Joel W Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research (JMLR)*, 2014.

[12] Dave Elliot. Flights of fancy: airborne wind turbines, 2014.

[13] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 2004.

[14] Johannes Kirschner and Andreas Krause. Information directed sampling and bandits with heteroscedastic noise. *Conference On Learning Theory (COLT)*, 2018.

[15] Johannes Kirschner, Ilija Bogunovic, Stefanie Jegelka, and Andreas Krause. Distributionally robust Bayesian optimization. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[16] Andreas Krause and Cheng Soon Ong. Contextual Gaussian process bandit optimization. *Conference on Neural Information Processing Systems (NeurIPS)*, 2011.

[17] Jinkyoo Park. Contextual Bayesian optimization with trust region (cbotr) and its application to cooperative wind farm control in region 2. *Sustainable Energy Technologies and Assessments*, 2020.

[18] C. E. Rasmussen and C. K. Williams. Gaussian processes for machine learning. *volume 1. MIT press Cambridge*, 2006.

[19] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *International Conference on Machine Learning (ICML)*, 2010.

Supplementary Material
Movement Penalized Bayesian Optimization with Application to Wind Energy Systems

Appendix A. Metrical Task Systems (MTS)

Let $(\mathcal{X}, d)$ be a finite metric space with $|\mathcal{X}| = n > 1$ as defined in Section 2. Henceforth, we denote the points in $\mathcal{X}$ as $\{x^1, \ldots, x^n\}$. The MTS problem runs over a single episode of horizon T. At every time instant $1 \leq t \leq T$, the learner receives a non-negative cost function over $\mathcal{X}$, $c_t : \mathcal{X} \to \mathbb{R}_+$ corresponding to each point in $\mathcal{X}$. The goal of any online optimization algorithm in this setting is to choose $x_t \in \mathcal{X}$ such that both the cost incurred over the rounds and sum over the distances between its subsequent decisions as measured by $d(x_t, x_{t-1})$ is minimized. We do not include episode m in the variables' subscripts as it runs over a single episode and instead include the time horizon T. Hence, $D_T = \{x_1, x_2, \ldots, x_T\}$ denotes the action sequence of length T outputted by an algorithm and $S_T(D_T)$ and $M_T(D_T)$ (Eq. (2)) denote the corresponding service and movement costs respectively. Then the total cost incurred by such an algorithm for initial state x_0 up to a time horizon T is

$$\text{cost}_T(D_T) = \sum_{t=1}^{T} c_t(x_t) + d(x_t, x_{t-1}).$$

Next, we recall some notions about competitive ratio for MTS from [9] which are useful to prove our regret guarantees in Appendix E. Here, D_T^* denotes the offline optimal sequence which minimizes $\text{cost}_T(D_T)$. Here we note that this offline optimal sequence D_T^* depends on the initial state x_0.

Competitive Ratio: If there exist constants α, β such that for every cost sequence $(c_t)_{t=1}^T$, arbitrary initial state $x_0 \in \mathcal{X}$ and distance metric $d(\cdot, \cdot)$,

$$\text{cost}_T(D_T) \quad \leq \quad \alpha \text{cost}_T(D_T^*) + \beta,$$

then the algorithm is α-competitive.

Refined Competitive Ratio Guarantees: If there exist constants $\alpha, \alpha', \beta, \beta'$ such that for every cost sequence $(c_t)_{t=1}^T$, arbitrary initial state $x_0 \in \mathcal{X}$ and distance metric $d(\cdot, \cdot)$,

$$S_T(D_T) \leq \alpha \text{cost}_T(D_T^*) + \beta, \tag{12}$$
$$M_T(D_T) \leq \alpha' \text{cost}_T(D_T^*) + \beta', \tag{13}$$

then the algorithm is α-competitive for service costs and α'-competitive for movement costs.

Appendix B. Hierarchically Separated Tree (HST) Metric

We define the tree $\mathcal{T} = (V, E)$ with the root vertex being r and weight corresponding to each vertex $v \in V$ being w_v. Let $\mathcal{L}$ denote the set of leaves in this tree $\mathcal{T}$. Consider the case when the weights are non-increasing while moving from root to any leaf. We assign the edge from any vertex u to its parent $par(u)$ the weight w_u.

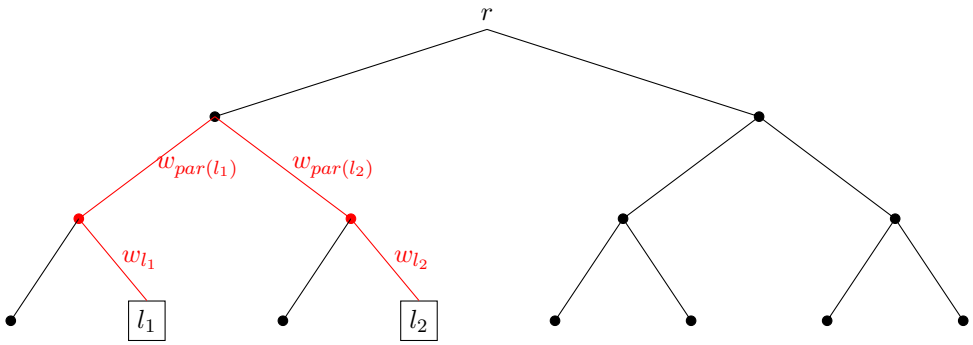


Figure 2: $d_{\mathcal{T}}(l_1, l_2) = w_{l_1} + w_{par(l_1)} + w_{l_2} + w_{par(l_2)}$.

Distance Function: We define the distance metric $d_{\mathcal{T}}(l, l')$ between any two leaves l, l' in the tree as the weighted length of the path from l to l'. For example, as shown in Fig. 2, Then the space with states $\mathcal{L}$ and distance metric $d_{\mathcal{T}}$ is defined as the HST metric space denoted by $(\mathcal{L}, d_{\mathcal{T}})$ ([9]).

B.1 τ-HST Metric

In an HST Metric space if the weights are exponentially decreasing, i.e., $w_u \leq w_{par(u)}/\tau$ then we call it a τ-HST Metric. Such metric spaces are of particular interest due to a result from [3] which states that any online algorithm which is $\mathcal{O}(g(n))$-competitive for τ-HST metric space is $\mathcal{O}(g(n)\log n)$-competitive for arbitrary n-point metric spaces where $g(n)$ is some function on n.

B.2 FRT Algorithm

The FRT algorithm from [13] is a randomized algorithm and outputs a tree $\mathcal{T}$ whose leaves correspond to points in $\mathcal{X}$ but the tree distance between any two points (leaves) $x^i, x^j \in \mathcal{X}$ (or $\mathcal{L}$) is $d_{\mathcal{T}}(x^i, x^j)$ and satisfies for any $x^i, x^j \in \mathcal{X}$

$$\mathbb{P}[d_{\mathcal{T}}(x^i, x^j) \geq d(x^i, x^j)] = 1, \tag{14}$$

$$\mathbb{E}_{\mathcal{T}}\left[d_{\mathcal{T}}(x^i, x^j)\right] \leq \mathcal{O}(\log n) d(x^i, x^j), \tag{15}$$

where $d(\cdot, \cdot)$, is the original metric and expectation is w.r.t. the random tree $\mathcal{T}$ generated by the FRT algorithm [13].

Appendix C. Action Representation

This section is intended to explain in detail about the randomized algorithm setup for MTS as elucidated in [9] and [6]. But instead of using the cost sequence $(c_t)_{t=1}^T$, we explain in terms of $f(\cdot, e_t)$ which is more relevant to our setting. In our proof, we only consider episodic expectation (see Eq. (31)) and want to understand how the randomized algorithm evolves within a single episode. Hence similar to MTS setup from Coester and Lee [9] we detail this randomization section for a single episode.

C.1 Action Randomization

Let $\mathcal{P}(\mathcal{X})$ be the set of probability measures supported on $\mathcal{X}$. For $\mu, \nu \in \mathcal{P}(\mathcal{X})$ denote $\mathbb{W}^1(\mu, \nu)$ as the Wasserstein-1-distance between μ and ν defined based on $d(\cdot, \cdot)$. A randomized online algorithm at time t outputs a random action $x_t \sim p_t$ where probability distribution $p_t \in \mathcal{P}(\mathcal{X})$. As defined earlier, we denote this random output sequence as $D_T = \{x_1, x_2, \ldots, x_T\}$. In this randomized setting it is more intuitive to consider the *expected cost*.

The *expected movement cost* for distance metric $d(\cdot, \cdot)$ is defined as follows:

$$\mathbb{E}\left[M_T(D_T)\right] = \mathbb{E}\left[\sum_{t=1}^{T} d(x_{t-1}, x_t)\right] = \sum_{t=1}^{T} \mathbb{E}[d(x_{t-1}, x_t)].$$

Here note that each term in the sum $\mathbb{E}[d(x_{t-1}, x_t)]$ is expectation w.r.t. a joint distribution of (x_{t-1}, x_t). Under certain assumptions on the sampling process (described in Appendix C.2), the joint distribution becomes

$$\mathbb{E}\left[M_T(D_T)\right] = \sum_{t=1}^{T} \mathbb{W}^1(p_{t-1}, p_t), \tag{16}$$

And the *expected service cost* is then,

$$\mathbb{E}\left[S_T(D_T)\right] = \mathbb{E}\left[\sum_{t=1}^{T} f(x_t, e_t)\right] = \sum_{t=1}^{T} \langle f(\cdot, e_t), p_t \rangle, \tag{17}$$

$$\mathbb{E}\left[\mathrm{cost}_T(D_T)\right] = \sum_{t=1}^{T} \mathbb{W}^1(p_{t-1}, p_t) + \sum_{t=1}^{T} \langle f(\cdot, e_t), p_t \rangle. \tag{18}$$

An randomized online algorithm is said to be α-*competitive* if for all $x_0 \in \mathcal{X}$ and any f, it outputs a sequence D_T whose expected cost for some $\beta > 0$ satisfies the following condition w.r.t. the offline optimal sequence D_T^* for some metric $d(\cdot, \cdot)$:

$$\mathbb{E}\left[\mathrm{cost}_T(D_T)\right] \leq \alpha \mathrm{cost}_T(D_T^*) + \beta.$$

C.2 Sampling from Joint Distribution

Since our goal is to minimize the total cost we choose a joint distribution $\zeta_t \in \Pi(p_{t-1}, p_t)$ which minimizes the expected cost as follows:

$$\mathbb{E}_{\zeta_t}[d(x_{t-1}, x_t)] = \inf_{\zeta \in \Pi(p_{t-1}, p_t)} \mathbb{E}_{\zeta}[d(U_{t-1}, U_t)],$$

where $\Pi(p_{t-1}, p_t)$ denotes the set of all random variables (U_{t-1}, U_t) whose marginals are p_{t-1} and p_t, respectively. By the definition of Wasserstein-1 distance where the Wasserstein cost function is assumed to be $d(\cdot, \cdot)$ we have,

$$\inf_{\zeta \in \Pi(p_{t-1}, p_t)} \mathbb{E}_{\zeta}[d(U_{t-1}, U_t)] = \mathbb{W}^1(p_{t-1}, p_t). \tag{19}$$

Hence, we want to ensure that the subsequent actions in algorithm 1 (x_{t-1}, x_t) follows the distribution ζ_t making our total movement cost

$$\mathbb{E}\left[M_T(D_T)\right] = \sum_{t=1}^{T} \mathbb{W}^1(p_{t-1}, p_t). \tag{20}$$

In order to achieve this, we take the following steps. We first note that the initial action x_0 is given. Hence, after obtaining p_1 using Mirror Descent procedure as done in Line-11 of Algorithm 1, we estimate ζ_1, then calculate the conditional distribution $\zeta_1(U_1|U_0 = x_0)$, and sample x_1 from this conditional distribution. At any future time instant t, after obtaining p_t, we repeat this process and calculate the conditional distribution $\zeta_t(U_t|U_{t-1} = x_{t-1})$ and sample x_t from it (Line 12 of Algorithm 1). In this way we ensure that at each time instant t, (x_{t-1}, x_t) is sampled from ζ_t and hence attaining the movement cost as sum of Wasserstein-1 distances.

C.3 Randomized Action Representation in τ-HST

In this section, we explain the randomized action representation in τ-HST space introduced in Section 3 in further detail. Also, we show the Wasserstein-1 distance (Eq. (19)) can be simplified in the $d_{\mathcal{T}}$-metric when the actions are leaves of $\mathcal{T}$ as done in [6] and [9].

From the analysis in the previous section, in order to calculate the movement cost, we need to compute the Wasserstein-1 distances between 2 probability distributions over the leaves of the constructed tree. As we now consider τ-HST metric space, the distance metric is $d_{\mathcal{T}}$ and Wasserstein-1 distance is denoted as $\mathbb{W}^1_{\mathcal{T}}(\cdot,\cdot)$. We first recall the following representation of the randomized action described in Section 3 from [6]. This will be useful in both the Wasserstein-1 distance calculation and in Algorithm 1.

Let $\mathcal{T} = (V, E)$ be a tree with vertices V, edges E, root r and leaves $\mathcal{L}$. We define a convex polytope on the space $\mathbb{R}^V_+$

$$K_{\mathcal{T}} := \Big\{ z \in \mathbb{R}_+^{|V|} : z_r = 1, z_u = \sum_{\nu \in \mathcal{C}(u)} z_\nu \quad \forall u \in V \backslash \mathcal{L} \Big\},$$

where $\mathcal{C}(u)$ denotes the children of u. Note that by the above definition for any $z \in K_{\mathcal{T}}$, it holds that

$$\sum_{l \in \mathcal{L}} z_l = 1.$$

And the Wasserstein-1 distance between 2 random actions in $(\mathcal{L}, d_{\mathcal{T}})$ specified by the probability distributions $l(z)$ and $l(z')$ is as follows:

$$\mathbb{W}^1_{\mathcal{T}}(l(z), l(z')) := \sum_{u \in V} w_u |z_u - z'_u| = ||z - z'||_{l_1(w)}. \tag{21}$$

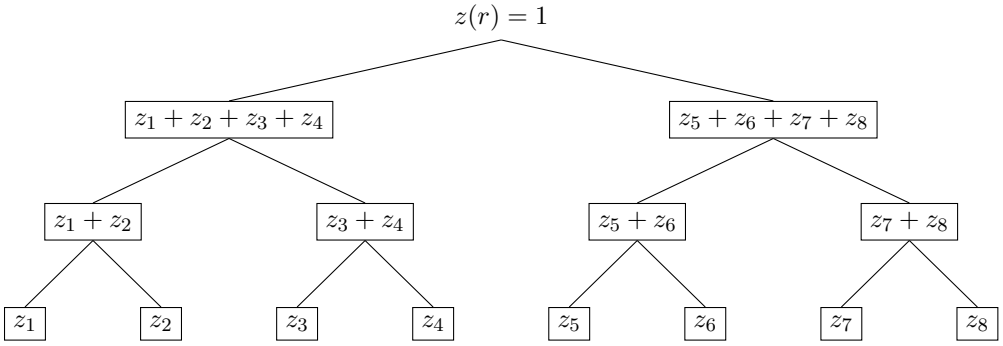
where w_u are the weights of the edges from u to $par(u)$ in $\mathcal{T}$. Hence the total cost w.r.t. $d_{\mathcal{T}}(\cdot,\cdot)$-metric denoted by $\text{cost}^{\mathcal{T}}_T(D_T)$ (Eq. (18)) now becomes

$$\mathbb{E}\left[\text{cost}^{\mathcal{T}}_T(D_T)\right] = \sum_{t=1}^{T} ||z_{t-1} - z_t||_{l_1(w)} + \sum_{t=1}^{T} \langle f(\cdot, e_t), l(z_t) \rangle. \tag{22}$$

Hence, for the τ-HST metric space $(\mathcal{L}, d_{\mathcal{T}})$, where leaves correspond to actions, z defines a probability distribution over all the states. Each entry z_u represents the probability that the selected action x belongs to the leaves of the subtree rooted at u, i.e., $z_u = \mathbb{P}(x \in \mathcal{L}(u))$. Also, We note that z is completely defined when all $z_l \in \mathcal{L}$ is provided. And for a deterministic state $x \in \mathcal{X}$, the corresponding state in $K_{\mathcal{T}}$ is

$$z_l = \left\{ \begin{array}{ll} 1, & \text{for } l = x \\ 0, & \text{for } l \neq x \end{array} \right\} \quad \forall l \in \mathcal{L}, \qquad z_u = \left\{ \begin{array}{ll} 1, & \text{for } x \in \mathcal{L}(u) \\ 0, & \text{for } x \notin \mathcal{L}(u) \end{array} \right\} \quad \forall u \in V \backslash \mathcal{L} \tag{23}$$

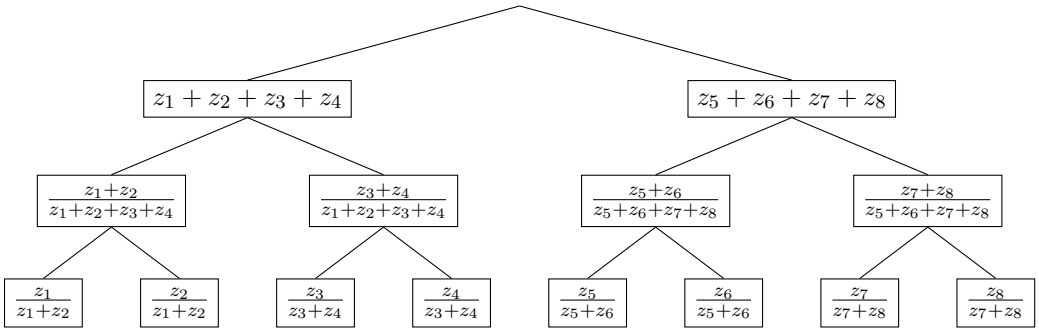
We can visualize this polytope with the example figure:



C.4 Action Representation using Conditional Probabilities

In Section 3, we provided an intutive understanding of how the algorithm works based on the calculation of $q^{(u)}$ corresponding to each internal vertex u (Eq. (10)). We also discussed how this can be viewed as a conditional probability $\mathbb{P}(x \in \mathcal{L}(\nu) | x \in \mathcal{L}(u))$ for $\nu \in \mathcal{C}(u)$ (children of u) and how it can be used to calculate the actual probabilities over the actions-$l(z)$ (Eq. (11)).

We first begin with a visualization of this q based on the visualization in Appendix C.3 and Eq. (11):



We explain why working in terms of q rather than z is beneficial. The Bregman divergence $D^{(u)}$ in the mirror descent update (Eq. (10)) uses a potential function and the authors of [9] observed that the conditional probability based potential function imitates the weighted entropy of the probability distribution over the leaves of a τ-HST tree. The intuition for the above statement from [9] is described next.

We first recall $Q_{\mathcal{T}}$ and $Q_{\mathcal{T}}^{(u)}$ from Section 3. $Q_{\mathcal{T}}$ is the set of valid conditional probabilities

$$Q_{\mathcal{T}} := \Big\{ q \in \mathcal{R}_{+}^{|V \setminus r|} : \sum_{\nu \in \mathcal{C}(u)} q_\nu = 1 \quad \forall u \in V \setminus \mathcal{L} \Big\}. \tag{24}$$

Moreover, given $q \in Q_{\mathcal{T}}$ we define the vector $q^{(u)} := [q_\nu,\ \nu \in \mathcal{C}(u)]$ as the conditional distribution over children of u, and let $Q_{\mathcal{T}}^{(u)}$ be the set of all valid distributions $q^{(u)}$.

We define the potential function $\Phi^{(u)}$ used for Bregman Divergence $D^{(u)}$ in the mirror descent update (Eq. (10)) for $q \in Q_{\mathcal{T}}$ and the corresponding $q^{(u)} \in Q_{\mathcal{T}}^{(u)}$ for a particular vertex $u \in V$ as follows:

$$\Phi^{(u)}(q^{(u)}) := \frac{1}{\kappa} \sum_{\nu \in \mathcal{C}(u)} \frac{w_\nu}{\eta_\nu} (q_\nu^{(u)} + \delta_\nu) \log(q_\nu^{(u)} + \delta_\nu), \tag{25}$$

where

$$\begin{aligned} \theta_u &:= \frac{|\mathcal{L}(u)|}{|\mathcal{L}(par(u))|}, \\ \eta_u &:= 1 + \log(1/\theta_u), \\ \delta_u &:= \frac{\theta_u}{\eta_u}, \\ \kappa &\geq 1 \quad \text{(fixed constant for all } u\text{)}. \end{aligned}$$

Appendix D. Algorithm

In this section we first provide some insights from [10, Sectiom 2.1] to solve the optimization problem in Eq. (10). Then we proceed to practically illustrate the flow of the algorithm from leaves to root w.r.t. $q \in Q_{\mathcal{T}}$ calculations using Eq. (10).

D.1 Divergence and Optimization Calculations

In order to solve the optimization problem Eq. (10), we first need to calculate the Bregman Divergence $D^{(u)}$. The Bregman Divergence for some potential function $\Phi(\cdot)$ is defined as follows:

$$D_\Phi(y||x) := \Phi(y) - \Phi(x) - \langle \nabla\Phi(x), y - x \rangle. \tag{26}$$

In our case, Bregman Divergence $D^{(u)}(\cdot||\cdot)$ calculates the divergence between two conditional probability vectors $q^{(u)}, q'^{(u)} \in Q_{\mathcal{T}}^{(u)}$ defined over the children of vertex u w.r.t. the potential function $\Phi^{(u)}$. Here $q, q' \in Q_{\mathcal{T}}$ (Eq. (24)) and potential function $\Phi^{(u)}$ is as defined in Eq. (25). Hence we have,

$$D^{(u)}(q^{(u)}||q'^{(u)}) := \frac{1}{\kappa} \sum_{\nu \in \mathcal{C}(u)} \frac{w_\nu}{\eta_\nu} \left[(q_\nu + \delta_\nu) \log(\frac{q_\nu + \delta_\nu}{q'_\nu + \delta_\nu}) + q'_\nu - q_\nu \right].$$

Now the authors of [9] use KKT conditions and Lagrange multipliers to solve Eq. (10) by substituting the definition of Bregman Divergence in Eq. (26). It yields that the solution to Eq. (10) for $q'^{(u)} = q_h^{(u)}$, $q^{(u)} = q_{h-1}^{(u)}$ and $e = e_{h,m}$ satisfies

$$\nabla\Phi^{(u)}(q'^{(u)}) = \nabla\Phi^{(u)}(q^{(u)}) - \mathrm{lcb}_m^{(u)}(\cdot, e) - \beta^{(u)} - \alpha^{(u)}. \tag{27}$$

Here $\beta^{(u)}$ and $\alpha^{(u)}$ are the Lagrange multipliers for the constraints in $Q_{\mathcal{T}}^{(u)}$ to ensure that for any $q^{(u)} \in Q_{\mathcal{T}}^{(u)}$, $q^{(u)}$ is actually a probability vector and comprises of the following 2 constraints,

$$\sum_{\nu \in \mathcal{C}(u)} q_\nu^{(u)} = 1 \qquad \text{and} \quad q_\nu^{(u)} \geq 0 \quad \text{for} \quad \nu \in \mathcal{C}(u).$$

Now calculating the gradient of the potential function in Eq. (25) we have,

$$\Big(\nabla \Phi^{(u)}(q^{(u)})\Big)_\nu = \frac{1}{\kappa} \frac{w_\nu}{\eta_\nu} (1 + \log(q_\nu + \delta_\nu)).$$

Substituting this in Eq. (27), the solution to Eq. (10) for $q'^{(u)} = q_h^{(u)}$, $q^{(u)} = q_{h-1}^{(u)}$ and $e = e_{h,m}$ will be

$$q_\nu'^{(u)} = (q_\nu^{(u)} + \delta_\nu) \exp\{\kappa \frac{\eta_\nu}{w_\nu} (\beta^{(u)} - (\mathrm{lcb}_m^{(u)}(\nu, e) - \alpha_\nu))\} - \delta_\nu. \tag{28}$$

Eq. (28) can be solved in polynomial time using interior point methods. But for practical purposes, we use projected gradient descent w.r.t. α to solve this problem.

Appendix E. Proof of Theorem 1

The first step in the proof of Theorem 1 is to rewrite the cumulative episodic regret (Eq. (4)) as a sum of expected episodic regret conditioned w.r.t. data observed till the previous episode. In particular, the main idea is to upper bound $R_{N_{ep}}^{\alpha,\beta} = \sum_{m=1}^{N_{ep}} r_m^{\alpha,\beta}$ by $\sum_{m=1}^{N_{ep}} \mathbb{E}_m[r_m^{\alpha,\beta} | \mathcal{F}_{m-1}]$. To do this, we make use of [14, Lemma 13] as we explain below. Here, the expectation $\mathbb{E}_m$ is w.r.t. the actions in episode m, $D_m = (x_{h,m})_{h=1}^H$ outputted by Algorithm 1, and $\mathcal{F}_m$ denotes the data collected by Algorithm 1 during the first m episodes, i.e.,

$$\mathcal{F}_m = \{(x_{h,i}, e_{h,i}, y_{h,i})_{h=1}^H\}_{i=1}^m. \tag{29}$$

We note that in the episodic regret

$$r_m^{\alpha,\beta} = \sum_{h=1}^H f(x_{h,m}, e_{h,m}) + \sum_{h=1}^H d(x_{h,m}, x_{h-1,m}) - \alpha \cdot \mathrm{cost}_m(D_m^*) - \beta,$$

the term $\alpha \cdot \mathrm{cost}_m(D_m^*) - \beta$ is constant w.r.t. the expectation $\mathbb{E}_m$ as it is independent of the actions $(x_{h,m})_{h=1}^H$. Also, the episodic cost $\mathrm{cost}_m(D_m) = \sum_{h=1}^H f(x_{h,m}, e_{h,m}) + \sum_{h=1}^H d(x_{h,m}, x_{h-1,m})$ is trivially upper bounded by $H(B + \psi)$. This is because $f(x, e) \leq B$ (follows from our assumptions $\|f\|_k \leq B$ and $k(\cdot, \cdot) \leq 1$) and $d(x, x') \leq \psi$ as detailed in Section 2.

Then, according to [14, Lemma 13], with probability at least $1 - \delta$ the cumulative regret from Eq. (4) is bounded as:

$$R_{N_{ep}}^{\alpha,\beta} = \sum_{m=1}^{N_{ep}} r_m^{\alpha,\beta} \tag{30}$$

$$\leq \sum_{m=1}^{N_{ep}} \mathbb{E}_m[r_m^{\alpha,\beta} | \mathcal{F}_{m-1}] + 4H(B + \psi) \log\Big(\frac{4\pi^2 N_{ep}^2}{3\delta}\big(\log(N_{ep}) + 1\big)\Big). \tag{31}$$

Above, we have applied [14, Lemma 13] to the cumulative sum of the stochastic process $r_m^{\alpha,\beta}$ and used the fact that each episodic cost $\text{cost}_m(D_m)$ is bounded by $0 \leq \text{cost}_m(D_m) \leq H(B+\psi)$.

In what follows, we focus on bounding

$$\sum_{m=1}^{N_{ep}} \mathbb{E}_m[r_m^{\alpha,\beta}|\mathcal{F}_{m-1}] = \sum_{m=1}^{N_{ep}} \mathbb{E}_m[\text{cost}_m(D_m) - \alpha \cdot \text{cost}_m(D_m^*) - \beta|\mathcal{F}_{m-1}]. \tag{32}$$

Recall that, $\mathbb{E}_m[\text{cost}_m(D_m)|\mathcal{F}_{m-1}] = \mathbb{E}_m[S_m(D_m)|\mathcal{F}_{m-1}] + \mathbb{E}_m[M_m(D_m)|\mathcal{F}_{m-1}]$. In the following sections we bound this sum of expected episodic costs using results from [9] and [11]. We begin by explicitly writing the episodic service costs in terms of the actions' distributions.

E.1 Costs in Terms of Conditional Distribution

Recall in Algorithm 1 the sequence of decisions $D_m = \{x_{1,m}, \ldots, x_{H,m}\}$ is sampled from conditional optimal coupling distribution denoted as $\zeta_{h-1,h,m}(U_{h,m} = x|U_{h-1,m} = x_{h-1,m})$ as stated in Algorithm 1, Line 13. Here $(U_{h-1,m}, U_{h,m})$ is a joint random variable whose marginal distributions are $l(z_{h-1,m})$ and $l(z_{h,m})$, respectively, and are computed as stated in Algorithm 1, Line 11.

Hence, the expected service cost of the algorithm w.r.t. $\text{lcb}_m(\cdot, e_{h,m})$ given $x_{h-1,m}$ becomes

$$\sum_{h=1}^{H} \mathbb{E}[\text{lcb}_m(x_{h,m}, e_{h,m})|x_{h-1,m}, \mathcal{F}_{m-1}] = \sum_{h=1}^{H} \langle \text{lcb}_m(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle.$$

Here the expectation $\mathbb{E}$ is w.r.t. the random variable $x_{h,m}$ whose probability distribution is $\zeta_{h-1,h,m}(U_{h,m} = x|U_{h-1,m} = x_{h-1,m})$ when conditioned on $x_{h-1,m}$. We note that the first state of each episode is sampled from $l(z_{1,m})$ (as there is no randomness in the initial state $x_{0,m}$ [2]). Also, within any given episode, $\text{lcb}_m(\cdot,\cdot)$ in GP-MD is fixed and does not get updated. Hence, by taking the total expectation over the whole episode and by using the law of total expectation [3], we arrive at:

$$\mathbb{E}_m\Big[\sum_{h=1}^{H} \langle \text{lcb}_m(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle|\mathcal{F}_{m-1}\Big] = \sum_{h=1}^{H} \langle \text{lcb}_m(\cdot, e_{h,m}), l(z_{h,m})\rangle. \tag{33}$$

On the other hand, the actual service cost for episode m is

$$S_m(D_m) = \sum_{h=1}^{H} f(x_{h,m}, e_{h,m}) \quad \text{where } x_{h,m} \sim \zeta_{h-1,h,m}(\cdot|x_{h-1,m}).$$

2. The optimal coupling conditional distribution minimizing Eq. (19) for $h = 1$ will be trivially satisfied by $l(z_{1,m})$ as the random variable $U_{0,m}$ is fixed at $x_{0,m}$

3. $\mathbb{E}_{x_{1,m},x_{2,m}}[\text{lcb}_m(x_{2,m}, e_{2,m})] = \sum_{y\in\mathcal{X}} \sum_{x\in\mathcal{X}} \text{lcb}_m(x, e_{h,m})\zeta_{1,2,m}(x|y)l(z_{1,m})(y)] = \sum_{x\in\mathcal{X}} \text{lcb}_m(x, e_{h,m}) \sum_{y\in\mathcal{X}} \zeta_{1,2,m}(x, y) = \sum_{x\in\mathcal{X}} \text{lcb}_m(x, e_{h,m})l(z_{2,m})(x)$ (can be similarly proved for any h using an inductive argument)

Hence, we can write

$$\mathbb{E}[f(x_{h,m}, e_{h,m})|x_{h-1,m}, \mathcal{F}_{m-1}] = \langle f(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle,$$

and

$$\sum_{h=1}^{H} \mathbb{E}[f(x_{h,m}, e_{h,m})|x_{h-1,m}, \mathcal{F}_{m-1}] = \sum_{h=1}^{H} \langle f(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle.$$

Moreover, conditioning on the event of Lemma 1, the cumulative cost above can be bounded in terms of its lower confidence bound as

$$\begin{aligned}\sum_{h=1}^{H} \langle f(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle \leq & \sum_{h=1}^{H} \langle \mathrm{lcb}_m(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle \\ & + \sum_{h=1}^{H} \langle 2\beta_m \sigma_m(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle.\end{aligned} \tag{34}$$

Also, by using the similar argument that led to Eq. (33), we have,

$$\mathbb{E}_m[S_m(D_m)|\mathcal{F}_{m-1}] = \mathbb{E}_m\Big[\sum_{h=1}^{H} \langle f(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle|\mathcal{F}_{m-1}\Big] = \sum_{h=1}^{H} \langle f(\cdot, e_{h,m}), l(z_{h,m})\rangle. \tag{35}$$

Then, combining Eq. (33), Eq. (34) and Eq. (35) we obtain

$$\begin{aligned}\sum_{h=1}^{H} \langle f(\cdot, e_{h,m}), l(z_{h,m})\rangle \leq & \sum_{h=1}^{H} \langle \mathrm{lcb}_m(\cdot, e_{h,m}), l(z_{h,m})\rangle \\ & + \mathbb{E}_m[\sum_{h=1}^{H} \langle 2\beta_m \sigma_m(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle|\mathcal{F}_{m-1}].\end{aligned} \tag{36}$$

By summing over all episodes (note that Lemma 1 holds uniformly over all episodes), we arrive at

$$\begin{aligned}\sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \langle f(\cdot, e_{h,m}), l(z_{h,m})\rangle \leq & \sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \langle \mathrm{lcb}_m(\cdot, e_{h,m}), l(z_{h,m})\rangle \\ & + \sum_{m=1}^{N_{ep}} \mathbb{E}_m[\sum_{h=1}^{H} \langle 2\beta_m \sigma_m(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle|\mathcal{F}_{m-1}].\end{aligned} \tag{37}$$

Now we define the offline optimal sequence for the cost sequence provided to Algorithm 1 in episode m $\{\mathrm{lcb}_m(\cdot, e_{1,m}), \ldots, \mathrm{lcb}_m(\cdot, e_{H,m})\}$ as $\{bx^*_{1,m}, \ldots, bx^*_{H,m}\}$ which will be useful in the analysis to bound $\sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \langle \mathrm{lcb}_m(\cdot, e_{h,m}), l(z_{h,m})\rangle$. Note that this offline optimal sequence is different from D^*_m as it is calculated for the cost sequence $\{\mathrm{lcb}_m(\cdot, e_{1,m}), \ldots, \mathrm{lcb}_m(\cdot, e_{H,m})\}$ and not $\{f(\cdot, e_{1,m}), \ldots, f(\cdot, e_{H,m})\}$.

In Algorithm 1, we use the mirror descent approach as proposed in [9] with input as the cost sequence $\{\text{lcb}_m(\cdot, e_{1,m}), \ldots, \text{lcb}_m(\cdot, e_{H,m})\}$ to output the actions D_m. Since Algorithm 1 uses $\text{lcb}_m(\cdot, e_{h,m})$ rather than $f(\cdot, e_{h,m})$ the guarantees provided in [9] for action sequence D_m will hold only w.r.t. $\{bx^*_{1,m}, \ldots, bx^*_{H,m}\}$ (the offline optimal sequence w.r.t. $\{\text{lcb}_m(\cdot, e_{1,m}), \ldots, \text{lcb}_m(\cdot, e_{H,m})\}$). Hence we can invoke Coester and Lee [9, Corollary 4], for the probability vector sequence $\{l(z_{1,m}), \ldots, l(z_{H,m})\}$ to guarantee that the sequence is 1-competitive in service costs (w.r.t. $\{\text{lcb}_m(\cdot, e_{1,m}), \ldots, \text{lcb}_m(\cdot, e_{H,m})\}$) and $\mathcal{O}((\log n)^2)-$competitive for movement costs w.r.t. the offline optimal sequence $\{bx^*_{1,m}, \ldots, bx^*_{H,m}\}$. Using the definition of refined competitive ratio guarantees Eq. (12) and Eq. (13) we have,

$$\sum_{h=1}^{H} \langle \text{lcb}_m(\cdot, e_{h,m}), l(z_{h,m}) \rangle \leq \mathcal{O}(1) + \sum_{h=1}^{H} \Big(\text{lcb}_m(bx^*_{h,m}, e_{h,m}) + d(bx^*_{h,m}, bx^*_{h-1,m}) \Big), \tag{38}$$

$$\mathbb{E}_m[M_m(D_m)|\mathcal{F}_{m-1}] \leq \mathcal{O}(1) + \mathcal{O}((\log n)^2) \sum_{h=1}^{H} \Big(\text{lcb}_m(bx^*_{h,m}, e_{h,m}) + d(bx^*_{h,m}, bx^*_{h-1,m}) \Big). \tag{39}$$

Focusing on $\sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \langle \text{lcb}_m(\cdot, e_{h,m}), l(z_{h,m}) \rangle$ in Eq. (37) and by using Eq. (38), we have

$$\sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \langle \text{lcb}_m(\cdot, e_{h,m}), l(z_{h,m}) \rangle \leq \sum_{m=1}^{N_{ep}} \Big(\mathcal{O}(1) + \sum_{h=1}^{H} \big(\text{lcb}_m(bx^*_{h,m}, e_{h,m}) + d(bx^*_{h,m}, bx^*_{h-1,m}) \big) \Big). \tag{40}$$

Since $D^*_m = \{x^*_{1,m}, \ldots, x^*_{H,m}\}$ is not optimal w.r.t. $\text{lcb}_m(\cdot, e_{h,m})$, it will incur more cost than $\{bx^*_{1,m}, \ldots, bx^*_{H,m}\}$ w.r.t. $\text{lcb}_m(\cdot, e_{h,m})$, i.e,

$$\sum_{h=1}^{H} \Big(\text{lcb}_m(bx^*_{h,m}, e_{h,m}) + d(bx^*_{h,m}, bx^*_{h-1,m}) \Big) \leq \sum_{h=1}^{H} \Big(\text{lcb}_m(x^*_{h,m}, e_{h,m}) + d(x^*_{h,m}, x^*_{h-1,m}) \Big). \tag{41}$$

Hence, by plugging Eq. (41) in Eq. (40) we have ,

$$\sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \langle \text{lcb}_m(\cdot, e_{h,m}), l(z_{h,m}) \rangle \leq \sum_{m=1}^{N_{ep}} \Big(\mathcal{O}(1) + \sum_{h=1}^{H} \big(\text{lcb}_m(x^*_{h,m}, e_{h,m}) + d(x^*_{h,m}, x^*_{h-1,m}) \big) \Big) \tag{42}$$

$$\leq \sum_{m=1}^{N_{ep}} \Big(\mathcal{O}(1) + \sum_{h=1}^{H} \big(f(x^*_{h,m}, e_{h,m}) + d(x^*_{h,m}, x^*_{h-1,m}) \big) \Big), \tag{43}$$

where the last equation Eq. (43) holds because we have conditioned on the event that confidence bounds hold true simultaneously for all episodes.

We can now use the above results to bound the cumulative service cost as

$$\sum_{m=1}^{N_{ep}} \mathbb{E}_m[S_m|\mathcal{F}_{m-1}] \overset{\text{(i)}}{=} \sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \langle f(\cdot, e_{h,m}), l(z_{h,m})\rangle \tag{44}$$

$$\overset{\text{(ii)}}{\leq} \sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \langle \text{lcb}_m(\cdot, e_{h,m}), l(z_{h,m})\rangle + \sum_{m=1}^{N_{ep}} \mathbb{E}_m[\sum_{h=1}^{H} \langle 2\beta_m \sigma_m(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle|\mathcal{F}_{m-1}]. \tag{45}$$

$$\overset{\text{(iii)}}{\leq} \sum_{m=1}^{N_{ep}} \Big(\mathcal{O}(1) + \sum_{h=1}^{H} \big(f(x^*_{h,m}, e_{h,m}) + d(x^*_{h,m}, x^*_{h-1,m})\big)\Big) + \sum_{m=1}^{N_{ep}} \mathbb{E}_m[\sum_{h=1}^{H} \langle 2\beta_m \sigma_m(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle|\mathcal{F}_{m-1}]. \tag{46}$$

Here (i) follows from Eq. (35), (ii) from Eq. (37) and (iii) from Eq. (43). Finally, we focus on bounding movement cost by using Eq. (39) and Eq. (43):

$$\sum_{m=1}^{N_{ep}} \mathbb{E}_m\Big[M_m(D_m)|\mathcal{F}_{m-1}\Big] = \sum_{m=1}^{N_{ep}} \mathbb{E}_m\Big[\sum_{h=1}^{H} d(x_{h,m}, x_{h-1,m})|\mathcal{F}_{m-1}\Big] \tag{47}$$

$$\leq \sum_{m=1}^{N_{ep}} \Big(\mathcal{O}(1) + \mathcal{O}((\log n)^2) \sum_{h=1}^{H} \big(\text{lcb}_m(bx^*_{h,m}, e_{h,m}) + d(bx^*_{h,m}, bx^*_{h-1,m})\big)\Big) \tag{48}$$

$$\leq \sum_{m=1}^{N_{ep}} \Big(\mathcal{O}(1) + \mathcal{O}((\log n)^2) \sum_{h=1}^{H} \big(f(x^*_{h,m}, e_{h,m}) + d(x^*_{h,m}, x^*_{h-1,m})\big)\Big). \tag{49}$$

E.2 Bounding Learning Error

As a last step, we focus on bounding the second term in Eq. (37) that we refer to as the *learning error*, i.e., $\sum_{m=1}^{N_{ep}} \mathbb{E}_m\Big[\sum_{h=1}^{H} \langle 2\beta_m \sigma_m(\cdot, e_{h,m}), \zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle|\mathcal{F}_{m-1}\Big]$. After that, we can finally bound the cumulative regret using the previously obtained bounds on service and movement costs (Eq. (46)and Eq. (49), respectively).

Consider the stochastic process,

$$\Delta_m = \sum_{h=1}^{H} \sigma_m^2(x_{h,m}, e_{h,m}).$$

Here $\sigma_m(\cdot, \cdot)$ is constructed from the data up to $m-1$ episodes defined earlier as $\mathcal{F}_{m-1}$. Now, using a similar argument that led to Eq. (33) (fixed $\sigma_m(\cdot, \cdot)$ for any given episode and law of

total expectation), the conditional mean of Δ_m given is $\mathcal{F}_{m-1}$

$$\mathbb{E}_m[\Delta_m|\mathcal{F}_{m-1}] = \sum_{h=1}^{H} \langle \sigma_m^2(\cdot, e_{h,m}), l(z_{h,m}) \rangle.$$

Now in order to bound this sum of conditional means of posterior variance $\sum_{m=1}^{N_{ep}} \mathbb{E}_m[\Delta_m|\mathcal{F}_{m-1}]$, by observed posterior variance $\sum_{m=1}^{N_{ep}} \Delta_m$ we use [14, Lemma 3]. Note that $\sigma_m^2(x, e) \leq 1$ by our assumption $k(\cdot, \cdot) \leq 1$ in Section 1 and the stochastic process Δ_m can be bounded as follows

$$\Delta_m = \sum_{h=1}^{H} \sigma_m^2(x_{h,m}, e_{h,m}) \leq \sum_{h=1}^{H} (1) \leq H.$$

Hence applying [14, Lemma 3] with probability at least $1 - \delta$,

$$\sum_{m=1}^{N_{ep}} \mathbb{E}_m[\Delta_m|\mathcal{F}_{m-1}] \leq 2\Big(\sum_{m=1}^{N_{ep}} (\Delta_m) \Big) + 4H \log(1/\delta) + 8H \log(4H) + 1,$$

This implies,

$$\sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \langle \sigma_m^2(\cdot, e_{h,m}), l(z_{h,m}) \rangle \leq 2 \sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \sigma_m^2(x_{h,m}, e_{h,m}) + 4H \log(1/\delta) + 8H \log(4H) + 1. \tag{50}$$

Now note that Eq. (50) cannot be directly bounded using bounds for sum of observed posterior variance as done in [19] and [8]. This is because $\sigma_m(\cdot, \cdot)$ is not updated continuously and is constant within any given episode m. Hence we first need to bound $\sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \sigma_m^2(x_{h,m}, e_{h,m})$ by $\sum_{m=1}^{N_{ep}} \sum_{h=1}^{H} \sigma_{h,m}^2(x_{h,m}, e_{h,m})$ where $\sigma_{h,m}(\cdot, \cdot)$ is constructed based on all $\{(x_{h,m}, e_{h,m}, y_{h,m})\}$ observed up to the round h in the m-th episode. To do this we use [11, Proposition 1], which bounds the ratio $\frac{\sigma_m(x,e)}{\sigma_{h,m}(x,e)}$ by the mutual information between $f(x, e)$ and observed function values upto round h in episode m ($\{y_{1,m}, \ldots, y_{h,m}\}$) conditioned on all function values observed upto episode $m-1$ ($\{(y_{h,i})_{h=1}\}_{i=1}^{m-1}$). For $I(\cdot; \cdot)$ denoting mutual information this can be written as follows,

$$\frac{\sigma_m(x, e)}{\sigma_{h,m}(x, e)} = \exp\big(I(f(x, e); y_{1,m:h,m}|y_{1,1:1,m})\big). \tag{51}$$

Further by Equations (11), (12), and (13) from [11], this conditional mutual information can be bounded as follows,

$$I(f(x, e); y_{1,m:h,m}|y_{1,1:1,m}) \leq \gamma_{H-1}. \tag{52}$$

Now, by using Eq. (51) and Eq. (52) we have

$$\sum_{m=1}^{N_{ep}}\sum_{h=1}^{H}\sigma_m^2(x_{h,m},e_{h,m}) \leq \sum_{m=1}^{N_{ep}}\Big(\exp(\gamma_{H-1})\big(\sum_{h=1}^{H}\sigma_{h,m}^2(x_{h,m},e_{h,m})\big)\Big) \tag{53}$$

$$\leq \exp(\gamma_H)\sum_{m=1}^{N_{ep}}\sum_{h=1}^{H}\sigma_{h,m}^2(x_{h,m},e_{h,m}) \tag{54}$$

$$\leq \exp(\gamma_H)\sum_{m=1}^{N_{ep}}\sum_{h=1}^{H}2\log(1+\sigma_{h,m}^2(x_{h,m},e_{h,m}) \tag{55}$$

$$\leq 2\exp(\gamma_H)\gamma_{N_{ep}H} \tag{56}$$

Here the last inequality follows from [8, Lemma 3]. Now, we are in position to focus on the learning error. It turns out that the learning error can be simplified by using the similar arguments as in Eq. (33), i.e.,

$$\sum_{m=1}^{N_{ep}}\mathbb{E}_m\Big[\sum_{h=1}^{H}\langle 2\beta_m\sigma_m(\cdot,e_{h,m}),\zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle\Big|\mathcal{F}_{m-1}\Big] = \sum_{m=1}^{N_{ep}}\sum_{h=1}^{H}2\beta_m\langle\sigma_m(\cdot,e_{h,m}),l(z_{h,m})\rangle. \tag{57}$$

As β_m is a non-decreasing sequence as stated in Lemma 1, we have

$$\sum_{m=1}^{N_{ep}}\sum_{h=1}^{H}2\beta_m\langle\sigma_m(\cdot,e_{h,m}),l(z_{h,m})\rangle \leq 2\beta_{N_{ep}}\sum_{m=1}^{N_{ep}}\sum_{h=1}^{H}\langle\sigma_m(\cdot,e_{h,m}),l(z_{h,m})\rangle \tag{58}$$

$$\leq 2\beta_{N_{ep}}\sqrt{N_{ep}H\sum_{m=1}^{N_{ep}}\sum_{h=1}^{H}(\langle\sigma_m(\cdot,e_{h,m}),l(z_{h,m})\rangle)^2} \tag{59}$$

$$\leq 2\beta_{N_{ep}}\sqrt{N_{ep}H\sum_{m=1}^{N_{ep}}\sum_{h=1}^{H}\langle\sigma_m^2(\cdot,e_{h,m}),l(z_{h,m})\rangle}. \tag{60}$$

We obtain Eq. (59) using Cauchy-Schwartz inequality and Eq. (60) using Jensen's inequality since $l(z_{h,m})$ is a probability distribution. Finally, the learning error will be bounded by

$$\begin{aligned}&\sum_{m=1}^{N_{ep}}\mathbb{E}_m\sum_{h=1}^{H}\langle 2\beta_m\sigma_m(\cdot,e_{h,m}),\zeta_{h-1,h,m}(\cdot|x_{h-1,m})\rangle|\mathcal{F}_{m-1}]\\ &\overset{\text{(i)}}{\leq} 2\beta_{N_{ep}}\sqrt{N_{ep}H\sum_{m=1}^{N_{ep}}\sum_{h=1}^{H}\langle\sigma_m^2(\cdot,e_{h,m}),l(z_{h,m})\rangle}\end{aligned} \tag{61}$$

$$\overset{\text{(ii)}}{\leq} 2\beta_{N_{ep}}\sqrt{2N_{ep}H\sum_{m=1}^{N_{ep}}\sum_{h=1}^{H}\sigma_m^2(x_{h,m},e_{h,m})+4H\log(1/\delta)+8H\log(4H)+1} \tag{62}$$

$$\overset{\text{(iii)}}{\leq} \mathcal{O}\Big(\beta_{N_{ep}}\sqrt{N_{ep}H(\exp(\gamma_H)\gamma_{N_{ep}H}+4H\log(1/\delta)+8H\log(4H)+1)}\Big). \tag{63}$$

Here (i) follows from Eq. (60), (ii) from Eq. (50) and (iii) from Eq. (56).

E.3 Bounding the regret

To bound the cumulative regret, recall our initial goal in Eq. (32) to bound

$$\sum_{m=1}^{N_{ep}} \mathbb{E}_m[\text{cost}_m(D_m)|\mathcal{F}_{m-1}] = \sum_{m=1}^{N_{ep}} \mathbb{E}_m[S_m(D_m) + M_m(D_m)|\mathcal{F}_{m-1}].$$

By using Eq. (63) in Eq. (46), we bound the service costs $\sum_{m=1}^{N_{ep}} \mathbb{E}_m[S_m(D_m)|\mathcal{F}_{m-1}]$ as

$$\sum_{m=1}^{N_{ep}} \mathbb{E}_m[S_m(D_m)|\mathcal{F}_{m-1}] \leq \sum_{m=1}^{N_{ep}} \Big(\mathcal{O}(1) + \sum_{h=1}^{H} \big(f(x^*_{h,m}, e_{h,m}) + d(x^*_{h,m}, x^*_{h-1,m})\big)\Big) + \mathcal{O}\Big(\beta_{N_{ep}} \sqrt{N_{ep}H(\exp(\gamma_H)\gamma_{N_{ep}H} + 4H\log(1/\delta) + 8H\log(4H) + 1)}\Big). \tag{64}$$

Also, Eq. (49) bounds movement costs as follows:

$$\sum_{m=1}^{N_{ep}} \mathbb{E}_m[M_m(D_m)|\mathcal{F}_{m-1}] \leq \sum_{m=1}^{N_{ep}} \Big(\mathcal{O}(1) + \mathcal{O}((\log n)^2) \sum_{h=1}^{H} \big(f(x^*_{h,m}, e_{h,m}) + d(x^*_{h,m}, x^*_{h-1,m})\big)\Big).$$

Combining this we obtain,

$$\sum_{m=1}^{N_{ep}} \mathbb{E}_m[\text{cost}_m(D_m)|\mathcal{F}_{m-1}] = \sum_{m=1}^{N_{ep}} \mathbb{E}_m[S_m(D_m) + M_m(D_m)|\mathcal{F}_{m-1}] \tag{65}$$

$$\leq \sum_{m=1}^{N_{ep}} \Big(\mathcal{O}(1) + \mathcal{O}((\log n)^2) \sum_{h=1}^{H} \big(f(x^*_{h,m}, e_{h,m}) + d(x^*_{h,m}, x^*_{h-1,m})\big)\Big) + \mathcal{O}\Big(\beta_{N_{ep}} \big(N_{ep}H(\exp(\gamma_H)\gamma_{N_{ep}H} + 4H\log(1/\delta) + 8H\log(4H))\big)^{\frac{1}{2}}\Big) \tag{66}$$

$$\leq \sum_{m=1}^{N_{ep}} \Big(\mathcal{O}(1) + \mathcal{O}((\log n)^2) \cdot \text{cost}_m(D^*_m)\Big) + \mathcal{O}\Big(\beta_{N_{ep}} \big(N_{ep}H(\exp(\gamma_H)\gamma_{N_{ep}H} + 4H\log(1/\delta) + 8H\log(4H))\big)^{\frac{1}{2}}\Big). \tag{67}$$

Thus, by setting set $\alpha = \mathcal{O}((\log n)^2)$ and $\beta = \mathcal{O}(1)$ the expected regret $\sum_{m=1}^{N_{ep}} \mathbb{E}_m[R_m^{\alpha,\beta}|\mathcal{F}_{m-1}]$ from Eq. (31) can be bounded using Eq. (67) as

$$\sum_{m=1}^{N_{ep}} \mathbb{E}_m[r_m^{\alpha,\beta}|\mathcal{F}_{m-1}] = \sum_{m=1}^{N_{ep}} \mathbb{E}_m[\text{cost}_m(D_m) - \mathcal{O}((\log n)^2) \cdot \text{cost}_m(D^*_m) - \mathcal{O}(1)|\mathcal{F}_{m-1}] \tag{68}$$

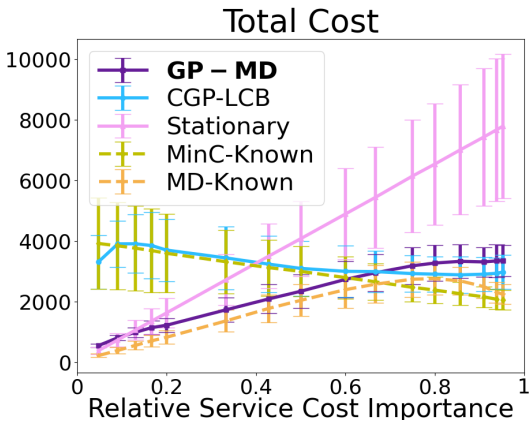
$$\leq \mathcal{O}\Big(\beta_{N_{ep}} \big(N_{ep}H(\exp(\gamma_H)\gamma_{N_{ep}H} + 4H\log(1/\delta) + 8H\log(4H))\big)^{\frac{1}{2}}\Big). \tag{69}$$

Now, for the final step of the regret guarantees is to ensure that Eq. (69) holds with probability at least $1-\delta$. For this, we take the union bound over the events such that Lemma 1, Eq. (50) and Eq. (31) hold. This effectively replaces δ by $\delta/3$ in each of the statements. Hence we get the required regret guarantee with probability at least $1-\delta$ as stated in Theorem 1:

$$R_{N_{ep}}^{\alpha,\beta} \leq \mathcal{O}\Big(\beta_{N_{ep}}\big(N_{ep}H\exp(\gamma_H)\gamma_{HN_{ep}} + H\log(\tfrac{H}{\delta})\big)^{\frac{1}{2}} + H(B+\psi)\log\big(\tfrac{N_{ep}\log(N_{ep})}{\delta}\big)\Big). \quad (70)$$

Appendix F. Synthetic Experiments

Synthetic experiments. Here the objective function is a random GP sample. The considered action space $\mathcal{X}$ is a subset of $[0,1]^2$ consisting of 400 points that form the uniform grid, while the context space $\mathcal{E}$ consists of 40 contexts that are uniformly sampled from $(0,1)$. We sample objective function (i.e., actual cost) $f : \mathcal{X} \times \mathcal{E} \to \mathbb{R}$ from a $GP(0,k)$, where k is a squared exponential kernel with lengthscale parameter set to $l = 0.2$. We use the Euclidean distance as the movement cost. Fig. 3a shows the algorithms' performance (for known kernel parameters) when run for 800 timesteps for varying importance of the service and movement costs by multiplying with $\rho/(1+\rho)$ and $1/(1+\rho)$, respectively. The performance of GP-MD is generally close to MD-KNOWN, which, as expected, performs the best. The stationary baseline performs comparably when ρ is small, while its performance deteriorates for larger values. As expected, both MINC-KNOWN and CGP-LCB incur higher total costs than GP-MD when the movement cost is of the higher or same relative importance as the service cost while the performance gap slowly decreases when the service cost becomes dominant. We observe that GP-MD 's performance is robust, i.e., it outperforms CGP-LCB whenever the movement cost dominates the total cost objective, while it remains comparable to CGP-LCB when the service cost dominates.



(a) Varying ρ

Figure 3: Total and movement cost performance of algorithms on synthetic functions for varying importance of movement/service cost (i.e., different ρ values). GP-MD outperforms CGP-LCB in terms of total incurred cost, and its performance closely follows one of the idealized benchmark MD-KNOWN. The performance of GP-MD remains robust when the movement cost importance in the total cost objective diminishes (Fig. 3a).