Bandit Convex Optimization for Scalable and Dynamic IoT Management

Tianyi Chen and Georgios B. Giannakis

***Abstract*—The present paper deals with online convex optimization involving both time-varying loss functions, and time-varying constraints. The loss functions are not fully accessible to the learner, and instead only the function values (a.k.a. bandit feedback) are revealed at queried points. The constraints are revealed after making decisions, and can be instantaneously violated, yet they must be satisfied in the long term. This setting fits nicely the emerging online network tasks such as fog computing in the Internet-of-Things (IoT), where online decisions must flexibly adapt to the changing user preferences (loss functions), and the temporally unpredictable availability of resources (constraints). Tailored for such human-in-the-loop systems where the loss functions are hard to model, a family of bandit online saddle-point (BanSaP) schemes are developed, which adaptively adjust the online operations based on (possibly multiple) bandit feedback of the loss functions, and the changing environment. Performance here is assessed by: i) *dynamic regret* that generalizes the widely used static regret; and, ii) *fit* that captures the accumulated amount of constraint violations. Specifically, BanSaP is proved to simultaneously yield sub-linear dynamic regret and fit, provided that the best dynamic solutions vary slowly over time. Numerical tests in fog computation offloading tasks corroborate that our proposed BanSaP approach offers competitive performance relative to existing approaches that are based on gradient feedback.**

***Index Terms*—Online learning, bandit convex optimization, saddle-point method, Internet of Things, mobile edge computing.**

I. Introduction

Internet-of-Things (IoT) envisions an intelligent infrastructure of networked smart devices offering task-specific monitoring and control services [1]. Leveraging advances in embedded systems, contemporary IoT devices are featured with *small-size* and *low-power* designs, but their computation and communication capabilities are limited. A prevalent solution during the past decade was to move computing, control, and storage resources to the remote cloud (a.k.a. data centers). Yet, the cloud-based IoT architecture is challenged by high latency due to directly communications with the cloud, which certainly prevents real-time applications [2]. Along with other features of IoT, such as *extreme heterogeneity* and *unpredictable dynamics*, the need arises for innovations in network design and management to allow for adaptive online service provisioning, subject to stringent delay constraints [3].

From the network design vantage point, *fog* is viewed as a promising architecture for IoT that distributes computation, communication, and storage closer to the end IoT users, along the cloud-to-things continuum [2]. In the fog computing paradigm, service provisioning starts at the network edge, e.g., smartphones, and high-tech routers, and only a portion of tasks will be offloaded to the powerful cloud for further processing (a.k.a. computation offloading) [4]–[6]. Existing approaches for computation offloading either focus on time-invariant static settings, or, rely on stochastic optimization approaches such as Lyapunov optimization to deal with time-varying cases; see [7] and references therein. Nevertheless, static settings cannot capture the changing IoT environment, and the stationarity commonly assumed in stochastic optimization literature may not hold in practice, especially when the stochastic process involves human participation as in IoT. From the management perspective, online network control, which is robust to non-stationary dynamics and amenable to *light-weight* implementations, remains a largely uncharted territory [5], [7].

Indeed, the *primary goal* of this paper is an algorithmic pursuit of online network optimization suitable for emerging tasks in IoT. Focusing on such algorithmic challenges, online convex optimization (OCO) is a promising methodology for sequential tasks with well-documented merits, especially when the sequence of convex costs varies in an unknown and possibly adversarial manner [8]. Aiming to empower traditional fog management policies with OCO, most available OCO works benchmark algorithms with a *static regret*, which measures the difference of costs (a.k.a. losses) between the online solution and the best static solution in hindsight [9], [10]. However, static regret is not a comprehensive performance metric in dynamic settings such as those encountered with IoT [11].

Recent works extend the analysis of static to that of *dynamic regret* [11], [12], but they deal with time-invariant constraints that cannot be violated instantaneously. Tailored for fog computing setups that need flexible adaptation of online decisions to dynamic resource availability, OCO with time-varying constraints was first studied in [13], along with its adaptive variant in [14], and the optimal regret bound in this setting was first established in [15]. Yet, the approaches in [13]–[15] remain operational under the premise that the loss functions are *explicitly known*, or, their gradients are readily available. Clearly, none of these two assumptions can be easily satisfied in IoT settings, because i) the loss function capturing user dissatisfaction, e.g., service latency or reliability, is hard to model in dynamic environments; and, ii) even if modeling is possible in theory, the low-power IoT devices may not afford the complexity of running statistical learning tools such as deep neural networks "on-the-fly."

In this context, targeting a gradient-free light-weight solution, alternative online schemes have been advocated leveraging point-wise values of loss functions (partial-information feedback) rather than their gradients (full-information feedback). They are termed bandit convex optimization (BCO) in machine learning [16]–[19], or referred as zeroth-order schemes

Work in this paper was supported by NSF 1509040, 1508993, and 1711471.

T. Chen and G. B. Giannakis are with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA. Emails: {chen3827, georgios}@umn.edu

TABLE I: A summary of related works on OCO/BCO

Reference	Benchmark	Constraints	Feedback
[8]–[10]	Static	Fixed and strict	Gradient
[11], [12]	Dynamic	Fixed and strict	Gradient
[15]	Static	Varying and long-term	Gradient
[13], [14]	Dynamic	Varying and long-term	Gradient
[22]	Static	Fixed and long-term	Grad./Fun. value
[16]–[21]	Static	Fixed and strict	Function value
This work	Dynamic	Varying and long-term	Function value

in optimization circles [20], [21]. While [16]–[18], [20] and [21] employed on BCO with time-invariant constraints that cannot be violated instantaneously, the *long-term* effect of such instantaneous violations was studied in [22], where the focus is still on static regret and time-invariant constraints. Building on full-information precursors [13]–[15], the present paper broadens the scope of BCO to the regime with *time-varying constraints*, and proposes a class of online algorithms termed online bandit saddle-point (BanSaP) approaches. With an eye on managing IoT with limited information, our contribution is the incorporation of long-term and time-varying constraints to expand the scope of BCO, as well as an improved regret-fit tradeoff relative to that in [22]; see a summary in Table I.

In a nutshell, relative to existing works, the main contributions of the present paper are summarized as follows.

c1) We generalize the standard BCO framework with only time-varying costs [16], [17], to account for both time-varying costs and constraints. Performance here is established relative to the best dynamic benchmark, via metrics that we term dynamic regret and fit (Section III).

c2) We develop a class of BanSaP algorithms to tackle this novel BCO problem, and analytically establish that BanSaP solvers yield simultaneously optimal sub-linear dynamic regret and fit, given that the accumulated variations of per-slot minimizers are known to grow sub-linearly with time (Section IV).

c3) Our BanSaP algorithms are applied to computation offloading tasks emerging in IoT management, and simulations demonstrate that the BanSaP solvers have comparable performance relative to full-information alternatives (Section V).

Notation. $(\cdot)^\top$ stands for vector and matrix transposition, and $\|\mathbf{x}\|$ denotes the ℓ_2-norm of a vector $\mathbf{x}$. Inequalities for vectors $\mathbf{x} > \mathbf{0}$, and the projection $[\mathbf{a}]^+ := \max\{\mathbf{a}, \mathbf{0}\}$ are entry-wise.

II. Bandit Online Learning with Constraints

In this section, a generic BCO formulation with long-term and time-varying constraints will be introduced, along with its real-world application in IoT management.

A. Online learning with constraints under partial feedback

Before introducing BCO with long-term constraints, we begin with the classical BCO setting, where constraints are time-invariant, and must be strictly satisfied [16], [17], [19]. Akin to its full-information counterpart [8], [9], BCO can be viewed as a repeated game between a learner and nature. Consider that time is discrete and indexed by t. Per slot t, a learner selects an action $\mathbf{x}_t$ from a convex set $\mathcal{X} \subseteq \mathbb{R}^d$, and subsequently nature chooses a loss function $f_t(\cdot) : \mathbb{R}^d \to \mathbb{R}$ through which the learner incurs a loss $f_t(\mathbf{x}_t)$. The convex feasible set $\mathcal{X}$ is a-priori known and fixed over the entire time horizon. Different from the OCO setup, at the end of each slot, only the value of $f_t(\mathbf{x}_t)$ rather than the form of $f_t(\mathbf{x})$ is revealed to the learner in BCO. Although this standard BCO setting is appealing to various applications such as online end-to-end routing [23] and task assignment [24], it does not account for potential variations of (possibly unknown) constraints, and does not deal with constraints that can possibly be satisfied in the long term rather than a slot-by-slot basis [13], [15], [22].

普通BCO缺点

Online optimization with time-varying and long-term constraints is well motivated for applications from power control in wireless communication [25], geographical load balancing in cloud networks [13], [26], to computation offloading in fog computing [27], [28]. Motivated by these dynamic network management tasks, our recent works [13], [14] studied OCO with time-varying constraints in *full information* setting, where the gradient feedback is available. Complementing [13] and [14], the present paper broadens the applicability of BCO to the regime with time-varying long-term constraints.

Specifically, we consider that per slot t, a learner selects an action $\mathbf{x}_t$ from a known and fixed convex set $\mathcal{X} \subseteq \mathbb{R}^d$, and then nature chooses not only a loss function $f_t(\cdot) : \mathbb{R}^d \to \mathbb{R}$, but also a time-varying penalty function $\mathbf{g}_t(\cdot) : \mathbb{R}^d \to \mathbb{R}^N$. The later gives rise to the time-varying constraint $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}$, which is driven by the unknown application-specific dynamics. Similar to the standard BCO setting, only the value of $f_t(\mathbf{x}_t)$ at the queried point $\mathbf{x}_t$ is revealed to the learner here; but different from the standard BCO setting, besides $\mathcal{X}$, the constraint $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}$ needs to be carefully taken care of. And the fact that $\mathbf{g}_t$ is unknown to the learner when performing her/his decision, makes it impossible to satisfy in every time slot. Hence, a more realistic goal here is to find a sequence of solutions $\{\mathbf{x}_t\}$ that minimizes the aggregate loss, and ensures that the constraints $\{\mathbf{g}_t(\mathbf{x}_t) \leq \mathbf{0}\}$ are satisfied in the long term on average. Specifically, extending the BCO framework [16]–[18] to accommodate such time-varying constraints, we consider the following online optimization problem

$$\min_{\{\mathbf{x}_t \in \mathcal{X}, \forall t\}} \sum_{t=1}^{T} f_t(\mathbf{x}_t) \quad \text{s. to} \quad \sum_{t=1}^{T} \mathbf{g}_t(\mathbf{x}_t) \leq \mathbf{0} \tag{1}$$

where T is the entire time horizon, $\mathbf{x}_t \in \mathbb{R}^d$ is the decision variable, f_t represents the cost function, $\mathbf{g}_t := [g_t^1, \ldots, g_t^N]^\top$ denotes the constraint function with nth entry $g_t^n(\cdot) : \mathbb{R}^d \to \mathbb{R}$, and $\mathcal{X} \in \mathbb{R}^d$ is a convex set. In the current setting, we assume that only the values of loss function are available at queried points since e.g., its complete form related to user experience is hard to approximate, but the constraint function is revealed to the learner as it represents measurable physical requirements e.g., power budget, and data flow conservation constraints. Before the algorithm development in Section III and performance analysis in Section IV, we will introduce a motivating example of fog computing in IoT.

B. Motivating setup: mobile fog computing in IoT

The online computational offloading task of fog computing in IoT [4], [5], [7] takes the form of BCO with long-term constraints (1). Consider a mobile network with a sensor layer, a fog layer, and a cloud layer [2], [3]. The sensor layer contains heterogeneous low-power IoT devices (e.g., wearable watches and smart cameras), which do not have enough computational capability, and usually offload their collected data to the local

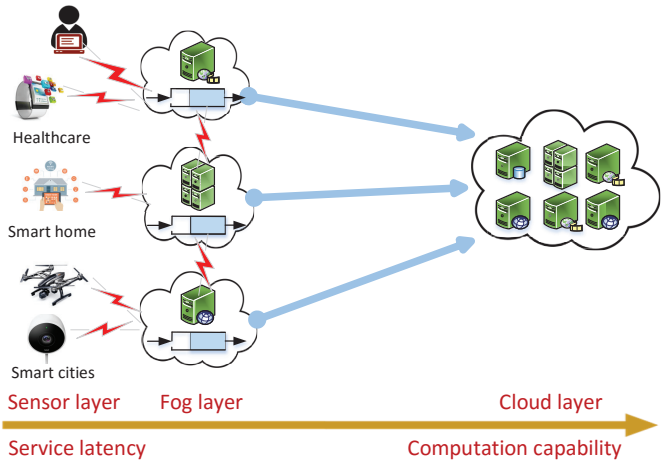

Fig. 1: A diagram of hierarchical fog computing framework.

fog nodes (e.g., smartphones and high-tech routers) in the fog layer for further processing [29]. The fog layer consists of N nodes in the set $\mathcal{N} := \{1, \ldots, N\}$ with moderate processing capability; thus, part of workloads will be collaboratively processed by the local fog servers to meet the stringent latency requirement, and the rest will be offloaded to the remote data center in the cloud layer [5]; also see Fig. 1.

Per time t, each fog node n collects data requests b_t^n from all its nearby sensors. Once receiving these requests, node n has *three options*: i) offloading the amount z_t^n to the remote data center; ii) offloading the amount y_t^{nk} to each of its nearby node k for collaborative computing; and, iii) locally processing the amount y_t^{nn} according to its resource availability. The optimization variable $\mathbf{x}_t$ in this case consists of the cloud offloading, local offloading, and local processing amounts; i.e., $\mathbf{x}_t := [z_t^1, \ldots, z_t^N, y_t^{11}, \ldots, y_t^{1N}, \ldots, y_t^{N1}, \ldots, y_t^{NN}]^\top$. Assuming that each fog node has a data queue to buffer unserved workloads, the instantaneously served workloads (offloading plus processing) is not necessarily equal to the data arrival rate. Instead, a long-term constraint is common to ensure that the cumulative amount of served workloads is no less than the arrived amount at each node n over time [25]

short term

$$\sum_{t=1}^{T} g_t^n(\mathbf{x}_t) := \sum_{t=1}^{T} \left(b_t^n + \sum_{k \in \mathcal{N}_n^{\text{in}}} y_t^{kn} - \sum_{k \in \mathcal{N}_n^{\text{out}}} y_t^{nk} - z_t^n - y_t^{nn} \right) \leq 0 \tag{2}$$

where $\mathcal{N}_n^{\text{in}}$ and $\mathcal{N}_n^{\text{out}}$ represent the sets of fog nodes with incoming links to node n and those with out-going links from node n, respectively. The bandwidth limit of communication link (e.g., wireline) from fog node n to the remote cloud is $\bar{z}^n$; the limit of the transmission link (e.g., wireless) from node n to its neighbor k is $\bar{y}^{nk}$, and the computation capability of node n is $\bar{y}^{nn}$. With $\bar{\mathbf{x}}$ collecting all the aforementioned limits, the feasible region can be expressed by $\mathbf{x}_t \in \mathcal{X} := \{\mathbf{0} \leq \mathbf{x}_t \leq \bar{\mathbf{x}}\}$.

Performance is assessed by the user dissatisfaction of the online processing and offloading decisions, e.g., aggregate delay [1], [3]. Specifically, as the computation delay is usually negligible for data centers with thousands of high-performance servers, the latency for cloud offloading amount z_t^n is mainly due to the communication delay, which is denoted as a time-varying cost $c_t^n(z_t^n)$ depending on the unpredictable network congestion during slot t. Likewise, the communication delay of the local offloading decision y_t^{nk} from node n to a nearby node k is denoted as $c_t^{nk}(y_t^{nk})$, but its magnitude is much lower than that of cloud offloading. Regarding the processing amount y_t^{nn}, its latency comes from the computation delay due to its limited computational capability, which is presented as a time-varying function $h_t^n(y_t^{nn})$ capturing the dynamic CPU capability during the computing processes. Per slot t, the network delay $f_t(\mathbf{x}_t)$ aggregates the computation delay at all nodes plus the communication delay at all links, namely

$$f_t(\mathbf{x}_t) := \sum_{n \in \mathcal{N}} \Big(\underbrace{c_t^n(z_t^n) + \textstyle\sum_{k \in \mathcal{N}_n^{\text{out}}} c_t^{nk}(y_t^{nk})}_{\text{communication}} + \underbrace{h_t^n(y_t^{nn})}_{\text{computation}} \Big). \tag{3}$$

Clearly, the explicit form of functions $c_t^n(\cdot)$, $c_t^{nk}(\cdot)$, and $h_t^n(\cdot)$ is *unknown* to the network operator due to the unpredictable traffic patterns [23]; but they are convex (thus $f_t(\mathbf{x}_t)$ is convex) with respect to their arguments, which implies that the marginal computation/communication latency is increasing as the offloading/processing amount grows.

Aiming to minimize the accumulated network delay while serving all the IoT workloads in the long term, the optimal offloading strategy in this mobile network is the solution of the following online optimization problem (cf. (3))

$$\min_{\{\mathbf{x}_t \in \mathcal{X}, \forall t\}} \sum_{t=1}^{T} f_t(\mathbf{x}_t), \text{ s. to } (2) \text{ for } n = 1, \ldots, N. \tag{4}$$

Comparing to the generic form (1), we consider an online fog computing problem in (4), where the loss (network latency) function $f_t(\cdot)$ and the data requests $\{b_t^n\}$ within slot t are not known when making the offloading and local processing decision $\mathbf{x}_t$; after performing $\mathbf{x}_t$, only the value of $f_t(\mathbf{x}_t)$ (a.k.a. loss) as well as the measurements $\{b_t^n\}$ are revealed to the network operator. In this example, measuring $\{b_t^n\}$ is tantamount to knowing the constraint function $g_t^n(\cdot)$ in (2). Therefore, (4) is in the form of (1).

III. Online Bandit Saddle-Point Methods

To solve the problem in Section II, an online saddle-point method is revisited first, before developing its bandit variants for network optimization with only partial feedback.

A. Online saddle-point approach with gradient feedback

Several works have studied the OCO setup with time-varying long-term constraints (cf. (1)), including [13], [15], and the recent variant [14] incorporating with adaptive stepsizes. Consider now the per-slot problem (1), which contains the current objective $f_t(\mathbf{x})$, the current constraint $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}$, and a time-invariant feasible set $\mathcal{X}$. With $\boldsymbol{\lambda} \in \mathbb{R}_+^N$ denoting the Lagrange multiplier associated with the time-varying constraint, the online Lagrangian of (1) can be expressed as

$$\mathcal{L}_t(\mathbf{x}, \boldsymbol{\lambda}) := f_t(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{g}_t(\mathbf{x}). \tag{5}$$

Serving as a basis for developing the bandit approaches, we next revisit the online saddle-point scheme with full-information [15], that is also equivalent to [13] when $\mathbf{g}_t(\mathbf{x})$ is linear. Specifically, given the primal iterate $\mathbf{x}_t$ and the dual iterate $\boldsymbol{\lambda}_t$ at each slot t, the next decision $\mathbf{x}_{t+1}$ is generated by

$$\mathbf{x}_{t+1} \in \arg\min_{\mathbf{x} \in \mathcal{X}} \nabla_{\mathbf{x}}^\top \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)(\mathbf{x} - \mathbf{x}_t) + \frac{1}{2\alpha} \|\mathbf{x} - \mathbf{x}_t\|^2 \tag{6}$$

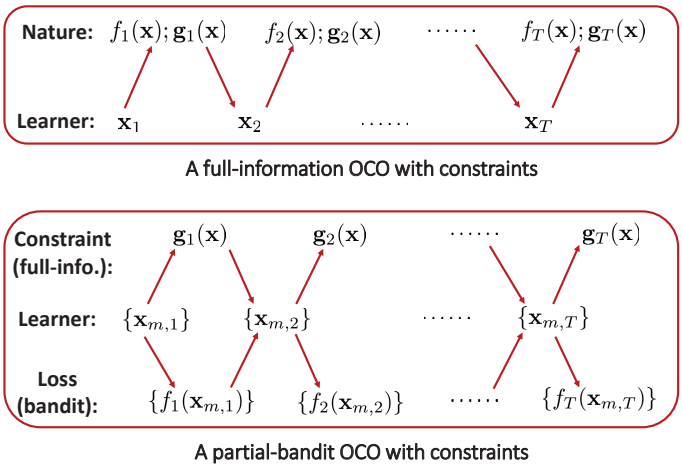

Fig. 2: A comparison of OCO with full/partial-bandit feedback.

where α is a pre-defined constant, and $\nabla_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) = \nabla f_t(\mathbf{x}_t) + \nabla^\top \mathbf{g}_t(\mathbf{x}_t)\boldsymbol{\lambda}_t$ is the gradient of $\mathcal{L}_t(\mathbf{x}, \boldsymbol{\lambda}_t)$ with respect to (w.r.t.) the primal variable $\mathbf{x}$ at $\mathbf{x} = \mathbf{x}_t$. The minimization (6) admits the closed-form solution, given by

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}(\mathbf{x}_t - \alpha\nabla_{\mathbf{x}}\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)) \tag{7}$$

where $\mathcal{P}_{\mathcal{X}}(\mathbf{y}) := \arg\min_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x} - \mathbf{y}\|^2$ denotes the projection operator. In addition, the dual update takes the modified online gradient ascent form

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \mu(\mathbf{g}_t(\mathbf{x}_t) + \nabla^\top \mathbf{g}_t(\mathbf{x}_t)(\mathbf{x}_{t+1} - \mathbf{x}_t))\right]^+ \tag{8}$$

where μ is a positive stepsize, and $\nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t) = \mathbf{g}_t(\mathbf{x}_t)$ is the gradient of $\mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda})$ w.r.t. $\boldsymbol{\lambda}$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}_t$. Note that (8) is a modified gradient update since the dual variable is updated along the first-order approximation of $\mathbf{g}_t(\mathbf{x}_{t+1})$ at the previous iterate $\mathbf{x}_t$ rather than $\mathbf{g}_t(\mathbf{x}_t)$ used in [13], which will be critical in our subsequent analytical derivations.

To perform the online saddle-point recursion (7)-(8) however, the gradient $\nabla f_t(\mathbf{x})$ and the constraint $\mathbf{g}_t(\mathbf{x})$ should be known to the learner at each slot t. When the gradient of $f_t(\mathbf{x})$ (or its explicit form) is unknown as it is in our setup, additional effort is needed. In this context, the systematic design of the online *bandit* saddle-point (BanSaP) methods will be leveraged to extend the online saddle-point method to the regime where gradient information is unavailable or computationally costly.

B. BanSaP with one-point partial feedback

The key idea behind BCO is to construct (possibly stochastic) gradient estimates using the limited *function value* information [16]–[18], [20], [21]. Depending on system variability, the online learner can afford one or multiple loss function evaluations (partial-information feedback) per time slot [17], [20], [21]. Intuitively, the performance of a bandit algorithm will improve if multiple evaluations are available per time slot; see Fig. 2 for a comparison of full- versus partial-information feedback settings.

To begin with, we consider the case where the learner can only observe the function value of $f_t(\mathbf{x})$ at a single point per slot t. The crux here is to construct a (possibly unbiased) estimate of the gradient using this single piece of feedback. Interestingly though, a stochastic gradient estimate of $f_t(\mathbf{x})$ can be obtained by one point *random* function evaluation [16]. The intuition can be readily revealed from the one-dimensional case ($d = 1$): For a binary random variable u taking values $\{-1, 1\}$ equiprobable, and a small constant $\delta > 0$, the idea of forward differentiation implies that the derivative f_t' at x can be approximated by

$$f_t'(x) \approx \frac{f_t(x+\delta) - f_t(x-\delta)}{2\delta} = \mathbb{E}_u\left[\frac{u}{\delta} f_t(x + \delta u)\right] \tag{9}$$

where the approximation is due to $\delta > 0$, and the equality follows from the definition of expectation. Hence, $f_t(x+\delta u)u/\delta$ can serve as a stochastic estimator of $f_t'(x)$ based only single function evaluation $f_t(x+\delta u)$. Generalizing this approximation to high dimensions, with a random vector $\mathbf{u}$ drawn from the unit sphere (a.k.a. the surface of a unit ball), the scaled function evaluation at a perturbed point $\mathbf{x} + \delta\mathbf{u}$ yields an estimate of the gradient $\nabla f_t(\mathbf{x})$, given by [16]

$$\nabla f_t(\mathbf{x}) \approx \mathbb{E}_{\mathbf{u}}\left[\frac{d}{\delta} f_t(\mathbf{x} + \delta\mathbf{u})\mathbf{u}\right] := \mathbb{E}_{\mathbf{u}}\left[\hat{\nabla}^1 f_t(\mathbf{x})\right] \tag{10}$$

where we define one-point gradient $\hat{\nabla}^1 f_t(\mathbf{x}) := \frac{d}{\delta} f_t(\mathbf{x}+\delta\mathbf{u})\mathbf{u}$.

Building upon this intuition, consider a bandit version of the online saddle-point iteration, for which the primal update becomes (cf. (7))

$$\hat{\mathbf{x}}_{t+1} = \mathcal{P}_{(1-\gamma)\mathcal{X}}\left(\hat{\mathbf{x}}_t - \alpha\hat{\nabla}^1_{\mathbf{x}}\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda}_t)\right) \tag{11}$$

where $(1-\gamma)\mathcal{X} := \{(1-\gamma)\mathbf{x} : \mathbf{x} \in \mathcal{X}\}$ is a subset of $\mathcal{X}$, $\gamma \in [0, 1)$ is a pre-selected constant depending on δ, and the one-point Langragian gradient is given by (cf. (10))

$$\hat{\nabla}^1_{\mathbf{x}}\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda}_t) := \hat{\nabla}^1 f_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)\boldsymbol{\lambda}_t. \tag{12}$$

In the full-information case, $\mathbf{x}_t$ in (7) is the learner's action, but in the bandit case the learner's action is $\mathbf{x}_{1,t} := \hat{\mathbf{x}}_t + \delta\mathbf{u}_t$, which is the point for function evaluation but not $\hat{\mathbf{x}}_t$ in (11). Furthermore, the projection is performed on a smaller convex set $(1-\gamma)\mathcal{X}$ in (11), which ensures feasibility of the perturbed $\mathbf{x}_{1,t} \in \mathcal{X}$. Similar to the full-information case (8), the dual update of BanSaP is given by

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \mu(\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t))\right]^+ \tag{13}$$

where μ is again the stepsize, and the learning iterate $\hat{\mathbf{x}}_t$ rather than the actual decision $\mathbf{x}_t$ is used in this update. Compared with the gradient-based recursions (7)-(8), the updates (11)-(13) with one-point bandit feedback do not increase computation or memory requirements, and thus provide a light-weight surrogate for gradient-free online bandit network optimization.

C. BanSaP with multipoint partial feedback

Featuring a simple update given minimal information, the BanSaP with one-point bandit feedback is suitable for fast-varying environments, where multiple function evaluations are impossible. As shown later in Sections IV and V, the theoretical and empirical performance of BanSaP with single-point evaluation is degraded relative to the full-information case.

To improve the performance of BanSaP with one-point feedback, we will first rely on two-point function evaluation at each slot [20], and then generalize to multipoint evaluation. Intuitively, this approach is justified when the underlying dynamics are slow, e.g., when the load and price profiles in

Algorithm 1 BanSaP for OCO with time-varying constraints

1: **Initialize:** primal iterate $\hat{\mathbf{x}}_1$, dual iterate $\boldsymbol{\lambda}_1$, parameters δ and γ, and proper stepsizes α and μ.
2: **for** $t = 1, 2 \ldots$ **do**
3: The learner plays the perturbed actions $\{\mathbf{x}_{m,t}\}_{m=1}^{M}$ based on the learning iterate $\hat{\mathbf{x}}_t$.
4: The nature reveals the losses $\{f_t(\mathbf{x}_{m,t})\}_{m=1}^{M}$ at queried points, and the constraint function $\mathbf{g}_t(\mathbf{x})$.
5: The learner updates the primal variable $\hat{\mathbf{x}}_{t+1}$ by (11) with the gradient estimated by (12) for $M = 1$, or, (15) for $M = 2$, otherwise, by (17).
6: The learner updates the dual variable $\boldsymbol{\lambda}_{t+1}$ via (13).
7: **end for**

power grids are piece-wise stationary. In this case, each slot can be further divided into multiple mini-slots, and one query is performed per mini-slot, over which the loss function and the constraints do not change. Compared to (11)-(13), the key difference is that the one-point estimate in (12) is replaced by

$$\hat{\nabla}^2 f_t(\hat{\mathbf{x}}_t) := \frac{d}{2\delta}\big(f_t(\hat{\mathbf{x}}_t + \delta\mathbf{u}_t) - f_t(\hat{\mathbf{x}}_t - \delta\mathbf{u}_t)\big)\mathbf{u}_t \tag{14}$$

where the function values are evaluated on two points around the learning iterate $\hat{\mathbf{x}}_t$, namely, $\mathbf{x}_{1,t} := \hat{\mathbf{x}}_t + \delta\mathbf{u}_t$ and $\mathbf{x}_{2,t} := \hat{\mathbf{x}}_t - \delta\mathbf{u}_t$ with $\mathbf{u}_t$ again drawn uniformly from the unit sphere $\mathbb{S} := \{\mathbf{u} \in \mathbb{R}^d : \|\mathbf{u}\| = 1\}$. The primal update becomes $\hat{\mathbf{x}}_{t+1} = \mathcal{P}_{(1-\gamma)\mathcal{X}}\big(\hat{\mathbf{x}}_t - \alpha\hat{\nabla}^2_{\mathbf{x}}\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda}_t)\big)$, with Lagrangian gradient

$$\hat{\nabla}^2_{\mathbf{x}}\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda}_t) := \hat{\nabla}^2 f_t(\hat{\mathbf{x}}_t) + \nabla\mathbf{g}_t(\hat{\mathbf{x}}_t)^\top\boldsymbol{\lambda}_t. \tag{15}$$

Similar to the one-point case, it is instructive to consider the two-point gradient estimate in the one-dimensional case ($d = 1$), where the expectation of the differentiation term in (14) approximates well the derivative of f_t at $\hat{x}_t$; that is,

$$\begin{aligned}&\mathbb{E}_u\left[\frac{u_t}{2\delta}\left(f_t(\hat{x}_t + \delta u_t) - f_t(\hat{x}_t - \delta u_t)\right)\right]\\ =&\frac{1}{2\delta}\left(f_t(\hat{x}_t + \delta) - f_t(\hat{x}_t - \delta)\right) \approx f_t'(\hat{x}_t)\end{aligned} \tag{16}$$

where the equality follows because the random variable u_t takes values $\{-1, 1\}$ equiprobable.

Relative to the one-point feedback case, the advantage of the two-point feedback is variance reduction in the gradient estimator. Specifically, the second moment of the stochastic gradient can be uniformly bounded, $\mathbb{E}[\|\frac{d}{2\delta}\big(f_t(\hat{\mathbf{x}}_t + \delta\mathbf{u}_t) - f_t(\hat{\mathbf{x}}_t - \delta\mathbf{u}_t)\big)\mathbf{u}_t\|^2] \leq d^2G^2$, where G is the Lipschitz constant of $f_t(\mathbf{x})$. This is in contrast to the one-point feedback where the second moment is inversely proportional to δ, since $\mathbb{E}[\frac{d}{\delta}\|f_t(\hat{\mathbf{x}}_t + \delta\mathbf{u}_t)\mathbf{u}_t\|^2] \leq d^2F^2/\delta^2$, with F denoting an upper-bound of $f_t(\mathbf{x})$. The proof of this argument can be found in the Appendix (Lemma 2). In fact, a bias-variance tradeoff emerges in the one-point case, but not in the two-point case. This subtle yet critical difference will be responsible for an improved performance of BanSaP with two-point feedback, and its stable empirical performance, as will be seen later.

With the insights gained so far, the next step is to endow the BanSaP with more than two function evaluations [17]. With $M > 2$ points, the gradient estimator is obtained by querying the function values over M points in the neighborhood of $\hat{\mathbf{x}}_t$. These points include $\mathbf{x}_{m,t} := \hat{\mathbf{x}}_t + \delta\mathbf{u}_{m,t}$, $1 \leq m \leq M-1$, and the learning iterate $\mathbf{x}_{m,t} := \hat{\mathbf{x}}_t$, where $\mathbf{u}_{m,t}$ is independently drawn from $\mathbb{S}$. Specifically, the gradient becomes (cf. (11))

$$\hat{\nabla}^M_{\mathbf{x}}\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda}_t) := \tag{17}$$
$$\frac{d}{\delta(M-1)}\sum_{m=1}^{M-1}\big(f_t(\hat{\mathbf{x}}_t + \delta\mathbf{u}_{m,t}) - f_t(\hat{\mathbf{x}}_t)\big)\mathbf{u}_{m,t} + \nabla\mathbf{g}_t(\hat{\mathbf{x}}_t)^\top\boldsymbol{\lambda}_t$$

where we define the M-point stochastic gradient as $\hat{\nabla}^M f_t(\hat{\mathbf{x}}_t) := \frac{d}{\delta(M-1)}\sum_{m=1}^{M-1}\big(f_t(\hat{\mathbf{x}}_t + \delta\mathbf{u}_{m,t}) - f_t(\hat{\mathbf{x}}_t)\big)\mathbf{u}_{m,t}$. At the price of extra computations, simulations will validate that the BanSaP with multipoint feedback enjoys improved performance. The family of the BanSaP approaches with one- or multiple-point feedback is summarized in Algorithm 1.

Remark 1 (Sampling schemes)**.** The BanSaP solvers here adopt uniform sampling for gradient estimation, meaning $\mathbf{u}$ is drawn uniformly from the unit sphere. However, other sampling rules can be incorporated without affecting the order of regret bounds derived later. For example, one can sample $\mathbf{u}$ from the canonical basis of a d-dimensional space uniformly at random [17], or, sample $\mathbf{u}$ from a normal distribution [21]. The effectiveness of these schemes will be tested using simulations.

IV. Performance analysis

In this section, we will introduce pertinent metrics to evaluate BanSaP algorithms in the online bandit learning with long-term constraints, and rigorously analyze the performance of the proposed algorithms.

A. Optimality and feasibility metrics

With regard to performance of BCO schemes, static regret is a common metric, under time-invariant and strictly satisfied constraints, which measures the difference between the aggregate loss and that of the best fixed solution in hindsight [16], [17]. Extending the definition of static regret to accommodate M-point function evaluations and time-varying constraints, let us first consider

$$\mathrm{Reg}^{\mathrm{s}}_T := \frac{1}{M}\sum_{t=1}^{T}\sum_{m=1}^{M}\mathbb{E}\left[f_t(\mathbf{x}_{m,t})\right] - \sum_{t=1}^{T} f_t(\mathbf{x}^*) \tag{18}$$

where the actual loss per slot is averaged over the losses of M actions (queried points), $\mathbb{E}$ is taken over the sequence of random actions (due to $\delta\mathbf{u}$ perturbations), and the best static solution is $\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\mathcal{X}}\sum_{t=1}^{T} f_t(\mathbf{x})$; s. to $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}$, $\forall t$. A BCO algorithm yielding a sub-linear regret implies that the algorithm is "on average" no-regret [22]; or, in other words, asymptotically not worse than the best fixed solution $\mathbf{x}^*$. Though widely used, the *static regret* relies on a rather coarse benchmark, which is not as useful in dynamic IoT settings. Specifically, the gap between the loss of the best static and that of the best dynamic benchmark is as large as $\mathcal{O}(T)$ [30].

In response to the quest for improved benchmarks in this dynamic setup with constraints, two metrics are considered here: *dynamic regret* and *dynamic fit*. The notion of dynamic regret has been recently adopted in [11], [12] to assess performance of online algorithms under time-invariant constraints. For our BCO setting of (1), we adopt

$$\mathrm{Reg}^{\mathrm{d}}_T := \frac{1}{M}\sum_{t=1}^{T}\sum_{m=1}^{M}\mathbb{E}\left[f_t(\mathbf{x}_{m,t})\right] - \sum_{t=1}^{T} f_t(\mathbf{x}^*_t) \tag{19}$$

where $\mathbb{E}$ is again taken over the sequence of random actions, and the benchmark is now formed via a sequence of best dynamic solutions $\{\mathbf{x}_t^*\}$ for the instantaneous cost minimization problem subject to the instantaneous constraint, namely

$$\mathbf{x}_t^* \in \arg\min_{\mathbf{x}\in\mathcal{X}} \; f_t(\mathbf{x}) \quad \text{s. to} \quad \mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}. \tag{20}$$

Quantitatively, the dynamic regret is always larger than the static regret, i.e., $\mathrm{Reg}_T^{\mathrm{s}} \leq \mathrm{Reg}_T^{\mathrm{d}}$, since $\sum_{t=1}^T f_t(\mathbf{x}^*)$ is always no smaller than $\sum_{t=1}^T f_t(\mathbf{x}_t^*)$ according to the definitions of $\mathbf{x}^*$ and $\mathbf{x}_t^*$. Hence, a sub-linear dynamic regret implies a sub-linear static regret, but not vice versa.

Regarding feasibility of decisions generated by a BCO algorithm, the notion of *dynamic fit* will be used to measure the accumulated violation of constraints [22], that is

$$\mathrm{Fit}_T^{\mathrm{d}} := \left\| \left[\frac{1}{M} \sum_{t=1}^{T} \sum_{m=1}^{M} \mathbf{g}_t(\mathbf{x}_{m,t}) \right]^+ \right\|. \tag{21}$$

Note that the dynamic fit is zero if the accumulated violation $\frac{1}{M}\sum_{t=1}^T\sum_{m=1}^M \mathbf{g}_t(\mathbf{x}_{m,t})$ is entry-wise less than zero. Hence, enforcing $\frac{1}{M}\sum_{t=1}^T\sum_{m=1}^M \mathbf{g}_t(\mathbf{x}_{m,t}) \leq \mathbf{0}$ is different from restricting $\mathbf{x}_t$ to meet $\frac{1}{M}\sum_{m=1}^M \mathbf{g}_t(\mathbf{x}_{m,t}) \leq \mathbf{0}$ in every slot. While the latter readily implies the former, the long-term constraint implicitly assumes that the instantaneous constraint violations can be compensated by the later strictly feasible decisions, and thus allows adaptation of online decisions to the unknown dynamics.

Under this broader BCO setup, an ideal online algorithm is the one that achieves both sub-linear dynamic regret and sub-linear dynamic fit. A sub-linear dynamic regret implies "no-regret" relative to the clairvoyant dynamic solution on the long-term average; i.e., $\lim_{T\to\infty} \mathrm{Reg}_T^{\mathrm{d}}/T = 0$; and a sub-linear dynamic fit indicates that the online strategy is also feasible on average; i.e., $\lim_{T\to\infty} \mathrm{Fit}_T^{\mathrm{d}}/T = 0$. Unfortunately, the sub-linear dynamic regret is not achievable in general, even when the time-varying constraint in (1) is absent [30]. Therefore, we aim at designing an online strategy that generates a sequence $\{\mathbf{x}_{m,t}\}$ ensuring sub-linear dynamic regret and fit, under the suitable conditions on the underlying dynamics.

B. Main results

Before formally analyzing the dynamic regret and fit for BanSaP, we assume that the following conditions are satisfied.

(as1) *For every t, the functions $f_t(\mathbf{x})$ and $\mathbf{g}_t(\mathbf{x})$ are convex.*

(as2) *Function $f_t(\mathbf{x})$ is bounded over the set $\mathcal{X}$, meaning $|f_t(\mathbf{x})| \leq F$, $\forall \mathbf{x} \in \mathcal{X}$; while $f_t(\mathbf{x})$ and $g_t^n(\mathbf{x})$ have bounded gradients; that is, $\|\nabla f_t(\mathbf{x})\| \leq G$, and $\max_n \|\nabla g_t^n(\mathbf{x})\| \leq G$.*

(as3) *For a small constant γ, there exists a constant $\eta > 0$, and an interior point $\tilde{\mathbf{x}} \in (1-\gamma)\mathcal{X}$ such that $\mathbf{g}_t(\tilde{\mathbf{x}}) \leq -\eta\mathbf{1}$, $\forall t$.*

(as4) *With $\mathbb{B} := \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq 1\}$ denoting the unit ball, there exist constants $0 < r \leq R$ such that $r\mathbb{B} \subseteq \mathcal{X} \subseteq R\mathbb{B}$.*

Assumptions (as1)-(as2) are typical in OCO with both full- and partial-information feedback [9], [16], [22]; (as3) is Slater's condition modified for our BCO setting, which guarantees the existence of a bounded Lagrange multiplier [31] in the constrained optimization; and, (as4) requires the action set to be bounded within a ball that contains the origin. When (as4) appears to be restrictive, it is tantamount to assuming $\mathcal{X}$ is compact and has a nonempty interior, because one can always apply an affine transformation (a.k.a. reshaping) on $\mathcal{X}$ to satisfy (as4); see [16, Section 3.2].

Under these assumptions, we are on track to first provide upper bounds for the dynamic regret, and the dynamic fit of the BanSaP solver with one-point feedback.

Theorem 1 (one-point feedback)**.** *Suppose that (as1)-(as4) are satisfied, and consider the parameters α, μ, δ, γ defined in* (11)-(13)*, and constants F, G, r, R defined in (as2)-(as4). If the dual variable is initialized by $\boldsymbol{\lambda}_1 = \mathbf{0}$, then the BanSaP with one-point feedback in* (7)-(8) *has dynamic regret bounded by*

$$\mathrm{Reg}_T^{\mathrm{d}} \leq \frac{R}{\alpha} V(\mathbf{x}_{1:T}^*) + \frac{R^2}{2\alpha} + \frac{d^2G^2R^2\alpha T}{\delta^2} + 2G\delta T + \gamma GRT\left(1 + \|\bar{\boldsymbol{\lambda}}\|\right) + 2\mu G^2R^2T \tag{22}$$

where $\|\bar{\boldsymbol{\lambda}}\| := \max_t \|\boldsymbol{\lambda}_t\|$, and the accumulated variation of the per-slot minimizers $\mathbf{x}_t^*$ in (20) *is given by*

$$V(\mathbf{x}_{1:T}^*) := \sum_{t=1}^{T} \|\mathbf{x}_t^* - \mathbf{x}_{t-1}^*\|. \tag{23}$$

In addition, the dynamic fit defined in (21) *is bounded by*

$$\mathrm{Fit}_T^{\mathrm{d}} \leq \frac{\|\bar{\boldsymbol{\lambda}}\|}{\mu} + \frac{G^2\sqrt{N}T}{2\beta} + \delta G\sqrt{N}T + \beta\sqrt{N}T\left(\frac{\alpha^2d^2F^2}{\delta^2} + \alpha^2G^2\|\bar{\boldsymbol{\lambda}}\|^2\right) \tag{24}$$

where $\beta > 0$ is a pre-selected constant. Furthermore, if we choose the stepsizes as $\alpha = \mu = \mathcal{O}(T^{-\frac{3}{4}})$, and the parameters $\delta = \mathcal{O}(T^{-\frac{1}{4}})$, $\beta = T^{\frac{1}{4}}$ and $\gamma = \delta/r$, then the online decisions generated by BanSaP are feasible, i.e., $\mathbf{x}_{1,t} \in \mathcal{X}$; and also yield the following dynamic regret and fit

$$\boxed{\mathrm{Reg}_T^{\mathrm{d}} = \mathcal{O}\left(V(\mathbf{x}_{1:T}^*)T^{\frac{3}{4}}\right) \quad \text{and} \quad \mathrm{Fit}_T^{\mathrm{d}} = \mathcal{O}\left(T^{\frac{3}{4}}\right).} \tag{25}$$

Proof: See Appendix B. ∎

For BanSaP with one-point feedback, Theorem 1 asserts that its dynamic regret and fit are upper-bounded by some constants depending on the those parameters, the time horizon, and the accumulated variation of per-slot minimizers. Interestingly, the crucial constant δ controlling the perturbation of random actions appears in both the denominator and numerator of (22) and (24), which correspond to the variance and bias of the gradient estimator. Therefore, simply setting a small δ will not only reduce the bias, but it will also boost the variance - a clear manifestation of the that is known as bias-variance tradeoff in BCO [18]. Optimally choosing parameters implies that the dynamic fit is sub-linearly growing, and the dynamic regret is sub-linear given that the variation of the per-slot minimizer is slow enough; i.e., $V(\mathbf{x}_{1:T}^*) = \mathbf{o}(T^{\frac{1}{4}})$.

Regarding BanSaP with two-point feedback, we can prove the following result that parallels Theorem 1.

Theorem 2 (two-point feedback)**.** *Consider the assumptions and the definitions of constants in Theorem 1. If the dual variable is initialized by $\boldsymbol{\lambda}_1 = \mathbf{0}$, then BanSaP with two-point feedback has dynamic regret bounded by*

$$\mathrm{Reg}_T^{\mathrm{d}} \leq \frac{R}{\alpha} V(\mathbf{x}_{1:T}^*) + \frac{R^2}{2\alpha} + 2\mu G^2R^2T + \alpha d^2G^2T + \gamma GRT(1 + \|\bar{\boldsymbol{\lambda}}\|) + 2\delta GT \tag{26}$$

and has dynamic fit in (21) *bounded by*

$$\mathrm{Fit}_T^{\mathrm{d}} \leq \frac{\|\bar{\boldsymbol{\lambda}}\|}{\mu} + \frac{G^2\sqrt{NT}}{2\beta} + \delta G\sqrt{NT} + \beta\sqrt{NT}\big(\alpha^2 d^2 G^2 + \alpha^2 G^2 \|\bar{\boldsymbol{\lambda}}\|^2\big). \quad (27)$$

In this case, if we choose the stepsizes as $\alpha = \mu = \mathcal{O}(T^{-\frac{1}{2}})$*, and set the parameters as* $\beta = T^{\frac{1}{2}}$*,* $\delta = \mathcal{O}(T^{-1})$*, and* $\gamma = \delta/r$*, then the online decisions generated by BanSaP are feasible, and its dynamic regret and fit are bounded by*

$$\boxed{\mathrm{Reg}_T^{\mathrm{d}} = \mathcal{O}\Big(V(\mathbf{x}_{1:T}^*)T^{\frac{1}{2}}\Big) \quad \text{and} \quad \mathrm{Fit}_T^{\mathrm{d}} = \mathcal{O}\big(T^{\frac{1}{2}}\big)} \quad (28)$$

where $V(\mathbf{x}_{1:T}^*)$ *is the accumulated variation of the per-slot minimizers* $\mathbf{x}_t^*$ *in* (23)*.*

Proof: See Appendix C. ∎

Comparing with the bounds in (22) and (24), the perturbation constant δ only appears in the numerator of (26) and (27) because our gradient estimator here replies on two points. In this case, the additional function evaluation allows BanSaP to choose an arbitrarily small δ to minimize the bias of stochastic gradient, without increasing its variance. This observation is aligned with those in BCO without long-term constraints [17], [18]. Furthermore, Theorem 2 establishes that the dynamic regret and fit are sub-linear if $V(\mathbf{x}_{1:T}^*) = \mathbf{o}(T^{\frac{1}{2}})$, which markedly improves those in Theorem 1 under one-point feedback.

For the case of BanSaP with $M > 2$ points, slightly improved bounds can be proved without changing the order of regret and fit, but they are omitted here for brevity. In addition, the bounds in Theorems 1 and 2 can be achieved without any knowledge of $V(\mathbf{x}_{1:T}^*)$. When the order of $V(\mathbf{x}_{1:T}^*)$ is known, or, can be estimated a-priori, tighter regret and fit bounds can be obtained by adjusting stepsizes accordingly. Formally, we can arrive at the following corollary.

Corollary 1. *Under the conditions of Theorems 1 and 2, suppose that there exists a constant* $\rho \in [0, 1)$ *such that the variation satisfies* $V(\mathbf{x}_{1:T}^*) = \mathbf{o}(T^{\rho})$*. If the stepsizes of BanSaP with one-point feedback are chosen as* $\alpha = \mu = \mathcal{O}\big(T^{\frac{3}{4}(\rho-1)}\big)$*, and the parameters are* $\delta = \mathcal{O}(T^{\frac{1}{4}(\rho-1)})$*,* $\beta = T^{\frac{3}{4}(1-\rho)}$*, and* $\gamma = \delta/r$*, then the dynamic regret and fit in* (25) *become*

$$\mathrm{Reg}_T^{\mathrm{d}} = \mathcal{O}\Big(T^{\frac{1}{4}(\rho+3)}\Big) \quad \text{and} \quad \mathrm{Fit}_T^{\mathrm{d}} = \mathcal{O}\left(T^{\frac{1}{4}(\rho+3)}\right). \quad (29)$$

Likewise, if the stepsizes of BanSaP with two-point feedback are chosen such that $\alpha = \mu = \mathcal{O}\big(T^{\frac{1}{2}(\rho-1)}\big)$*, and the parameters are* $\delta = \mathcal{O}(T^{\frac{1}{2}(\rho-1)})$*,* $\beta = T^{\frac{1}{2}(1-\rho)}$*, and* $\gamma = \delta/r$*, then the dynamic regret and fit in* (25) *become*

$$\mathrm{Reg}_T^{\mathrm{d}} = \mathcal{O}\Big(T^{\frac{1}{2}(\rho+1)}\Big) \quad \text{and} \quad \mathrm{Fit}_T^{\mathrm{d}} = \mathcal{O}\left(T^{\frac{1}{2}(\rho+1)}\right). \quad (30)$$

Apparently, Corollary 1 implies that sub-linear dynamic regret and fit are both possible, provided that the accumulated variation of the minimizers is growing sub-linearly ($\rho < 1$), and it is available to the learner in advance. It provides valuable insights for choosing optimal stepsizes in dynamic environments. Specifically, adjusting stepsizes to match the variability of the environment is the key to achieving the optimal dynamic regret and fit. Intuitively, when the variation is fast (large ρ), slowly decaying stepsizes (thus larger stepsizes) can better track the potential changes; and vice versa.

Algorithm 2 BanSaP for fog computation offloading

1: **Initialize:** primal iterates $\{\hat{y}_1^{nk}\}$ and $\{\hat{z}_1^n\}$, dual iterate $\boldsymbol{\lambda}_1$, parameters δ and γ, and proper stepsizes α and μ.
2: **for** $t = 1, 2\ldots$ **do**
3: **for** $m = 1, \ldots, M$ **do**
4: Fog nodes perform *perturbed* offloading decisions to cloud $\{z_{m,t}^n\}$, to neighbor edges $\{y_{m,t}^{nk}\}$, and locally process $\{y_{m,t}^{nn}\}$ based on $\hat{\mathbf{x}}_t$.
5: **end for**
6: Fog nodes observe the (possibly multiple) losses to update (31) with stochastic gradients obtained via (32).
7: Fog nodes observe the actual user demands from IoT devices to update the dual variables (33).
8: **end for**

Remark 2 (Optimal regret)**.** As a special case of Theorems 1 and 2, by confining $\mathbf{x}_1^* = \cdots = \mathbf{x}_T^*$ so that $V(\mathbf{x}_{1:T}^*) = 0$, the dynamic regret bounds (25) and (28) reduce to the static ones, which correspond to $\mathcal{O}(T^{\frac{3}{4}})$ in the one-point feedback case, and to $\mathcal{O}(\sqrt{T})$ in the two-point case. This pair of bounds markedly improves the *regret versus fit tradeoff* in [22], and matches the order of regret in [16], and [17], [20], which are the best possible ones that can be achieved by *efficient* algorithms even in the BCO setup without the long-term constraints.

Remark 3 (Dynamic regret)**.** Theorems 1, 2 and Corollary 1 extend the dynamic regret analysis in [11]–[13] to the regime of *bandit* online learning with long-term *time-varying* constraints. Interestingly though, in the BCO setting of our interest, sub-linear dynamic regret and fit are possible to achieve when the per-slot minimizer *does not vary on average*, that is, $V(\mathbf{x}_{1:T}^*)$ is sub-linearly growing with T.

V. Numerical Tests

In this section, we demonstrate how the fog computation offloading task can benefit from our novel BanSaP solvers.

A. BanSaP for fog computation offloading

Recall that the computation offloading problem (4) is in the form of (1). Therefore, the BanSaP solver of Section III can be customized to solve (4) in an *online* fashion, with provable performance and feasibility guarantees.

Specifically, with $\mathbf{g}_t(\mathbf{x}_t)$ as in (2) and $f_t(\mathbf{x}_t)$ as in (3), the primal update (7) boils down to a simple closed-form gradient update amenable to decentralized implementation; the cloud offloading amount at node n is

$$\hat{z}_{t+1}^n = \Big[\hat{z}_t^n - \alpha\big(\hat{\nabla} c_t^n(\hat{z}_t^n) - \lambda_t^n\big)\Big]_0^{\bar{z}^n} \quad (31a)$$

and the offloading amount from node n to node k is given by

$$\hat{y}_{t+1}^{nk} = \Big[\hat{y}_t^{nk} - \alpha\big(\hat{\nabla} c_t^{nk}(\hat{y}_t^{nk}) - \lambda_t^n + \lambda_t^k\big)\Big]_0^{\bar{y}^{nk}} \quad (31b)$$

while the local processing decision at node n is generated by

$$\hat{y}_{t+1}^{nn} = \Big[\hat{y}_t^{nn} - \alpha\big(\hat{\nabla} h_t^n(\hat{y}_t^{nn}) - \lambda_t^n\big)\Big]_0^{\bar{y}^{nn}} \quad (31c)$$

where α is chosen according to Theorems 1 and 2. Using two-point feedback ($M = 2$) as an example, the gradients involved in (31) can be estimated as

$$\hat{\nabla}^2 c_t^n(\hat{z}_t^n) := \frac{d}{2\delta}\Big(f_t(\underbrace{\hat{\mathbf{x}}_t + \delta \mathbf{u}_t}_{\mathbf{x}_{1,t}}) - f_t(\underbrace{\hat{\mathbf{x}}_t - \delta \mathbf{u}_t}_{\mathbf{x}_{2,t}})\Big) u_t(\hat{z}^n) \quad (32a)$$

and with respect to the offloading variable, as

$$\hat{\nabla}^2 c_t^{nk}(\hat{y}_t^{nk}) := \frac{d}{2\delta}\big(f_t(\hat{\mathbf{x}}_t + \delta \mathbf{u}_t) - f_t(\hat{\mathbf{x}}_t - \delta \mathbf{u}_t)\big) u_t(\hat{y}^{nk}) \quad (32b)$$

and with respect to the local processing variable, as

$$\hat{\nabla}^2 h_t^n(\hat{y}_t^{nn}) := \frac{d}{2\delta}\big(f_t(\hat{\mathbf{x}}_t + \delta \mathbf{u}_t) - f_t(\hat{\mathbf{x}}_t - \delta \mathbf{u}_t)\big) u_t(\hat{y}^{nn}) \quad (32c)$$

where $u_t(\hat{z}^n)$, $u_t(\hat{y}^{nk})$, and $u_t(\hat{y}^{nn})$ represent the corresponding entries of the random vector $\mathbf{u}_t \in \mathbb{R}^{|\mathcal{E}|}$ at slot t.

The dual update (8) at each node n reduces to

$$\lambda_{t+1}^n = \left[\lambda_t^n + \mu\left(b_t^n + \sum_{k\in\mathcal{N}_n^{\text{in}}} \hat{y}_{t+1}^{kn} - \sum_{k\in\mathcal{N}_n^{\text{out}}} \hat{y}_{t+1}^{nk} - \hat{z}_{t+1}^n - \hat{y}_{t+1}^{nn}\right)\right]^+ \quad (33)$$

where μ is chosen according to Theorems 1 and 2. Intuitively, to guarantee completion of the service requests, the dual variable increases (increasing penalty) when there is instantaneous service residual, and decreases when over-serving incurs in the mobile-edge computing systems. Following its generic form in Algorithm 1, BanSaP for online fog computation offloading tasks, is summarized in Algorithm 2.

B. Numerical experiments

Consider the fog computing task in Section II-B with $N = 10$ nodes and a cloud center. Each fog node has an outgoing link to the cloud, and two outgoing links to two nearby fog nodes for local collaborative computing. For a communication link offloading loads from node n to k, the offloading limit is $\bar{y}^{nk} = 10$, the local computation limit at node n is $\bar{y}^{nn} = 50$, and the fog-cloud offloading limits $\{\bar{z}^n\}$ are all set to 100. The online cost (a.k.a. service latency) in (3) is specified by

$$f_t(\mathbf{x}_t) := \sum_{n\in\mathcal{N}} \left(e^{p_t^n z_t^n} + \textstyle\sum_{k\in\mathcal{N}_n^{\text{out}}} l^{nk} y_t^{nk} + l^{nn}(y_t^{nn})^2\right) \quad (34)$$

where $p_t^n = 0.015\sin(\pi t/96) + 0.05$, $n \in \mathcal{N}\backslash\{4,5\}$, $p_t^n = 0.045\sin(\pi t/96) + 0.15$, $n \in \{4,5\}$, and the local coefficients are set to $l^{nk} = 8/\bar{y}^{nk}$ and $l^{nn} = 8/\bar{y}^{nn}$. Regarding the data arrival rate b_t^n, it is generated according to $b_t^n = q^n \sin(\pi t/96) + \nu_t^n$, with q^n and ν_t^n uniformly distributed over $[40, 50]$ and $[45, 55]$ for $n \in \mathcal{N}\backslash\{1,2,3\}\bigcup\{4,5\}$, and $q^n \in [32, 40], \nu_t^n \in [36, 44]$, $n \in \{1,2,3\}$, and $q^n \in [20, 25], \nu_t^n \in [22.5, 27.5]$, $n \in \{4,5\}$. Notice that the scales of p_t^n and b_t^n vary between nodes, mimicking heterogeneity of real IoT systems; and the periods of p_t^n and b_t^n correspond to a 24-hour interval with slot duration 7.5 minutes. When the parameters of BanSaP need to be slightly adjusted in each test, they are set to $\gamma = 0.05$, and $\delta = 4$ for with $M = 1$, and $\delta = 0.05$ for $M \geq 2$.

Finally, BanSaP is benchmarked by: i) the *full-information* MOSP method in [13] that takes gradient-based update for primal-dual variables; ii) the *heuristic* cloud-only approach that offloads all data requests to the remote cloud; and, iii) the *heuristic* fog-only approach that processes all data requests locally without collaboration. For both cloud-only and fog-only approaches, unoffloaded and unprocessed requests are buffered at the fog nodes for later processing; thus, these amounts are measured by their fit. As different stepsizes of BanSaP and MOSP lead to different behaviors, we manually optimized stepsizes in each test so that they have similar fit, and focus on their cost comparison. All simulated tests were averaged over 500 Monte Carlo realizations.

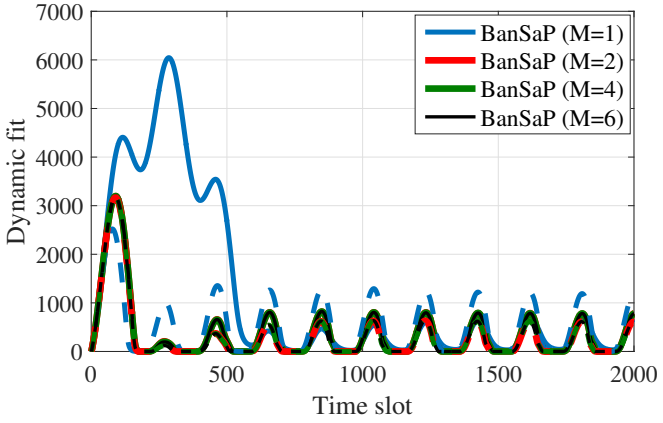

Fig. 3: Effect of sampling schemes and number of feedback on dynamic fit. Solid lines: BanSaP with uniformly sampling from a unit sphere (uniform sampling). Dashed lines: BanSaP with randomly sampling from standard basis (coordinate sampling).

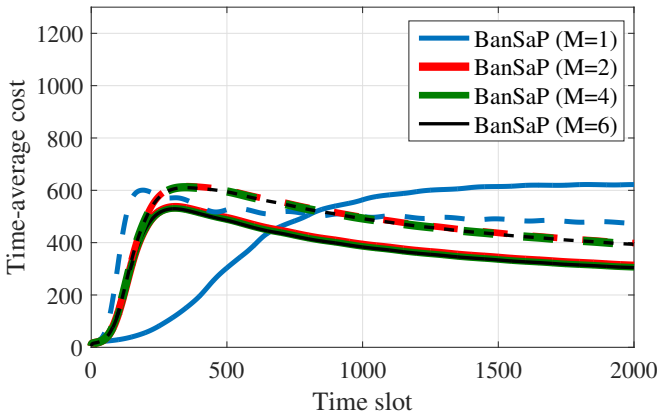

Fig. 4: Effect of sampling schemes and number of feedback on average cost. Solid lines: BanSaP with sampling from a unit sphere (uniform sampling). Dashed lines: BanSaP with randomly sampling from standard basis (coordinate sampling).

Effect of complexity and sampling schemes. In a simplified setting with $N = 5$ nodes, the fit and average cost are compared among the BanSaP variants with M-point feedback under different sampling schemes in Figs. 3 and 4. Clearly, for both sampling schemes, the cost and fit of BanSaP solvers decrease as the amount of bandit feedback increases. However, such performance gain varnishes when feedback increases; e.g., $M \geq 4$. Regarding the sampling schemes, Fig. 3 demonstrates that when all the BanSaP variants have low dynamic fit, the uniform sampling-based BanSaP with one-point feedback has large initial fit; and Fig. 4 confirms that for $M = 1$, the coordinate sampling-based BanSaP outperforms that with uniform sampling; and, for $M \geq 2$, the BanSaP solvers with uniform sampling incur lower cost. Therefore, to optimize empirical

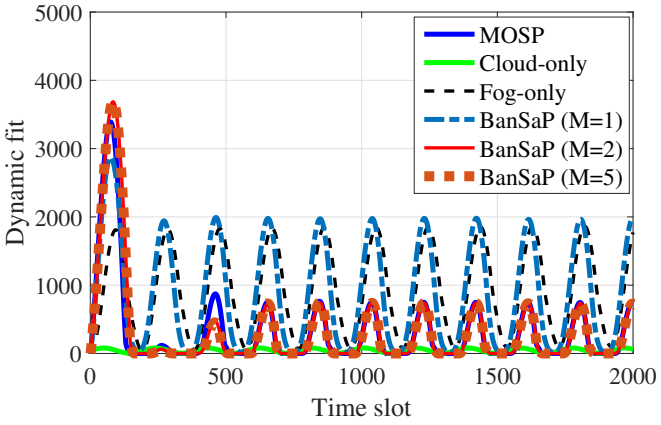

Fig. 5: Comparison based on dynamic fit.

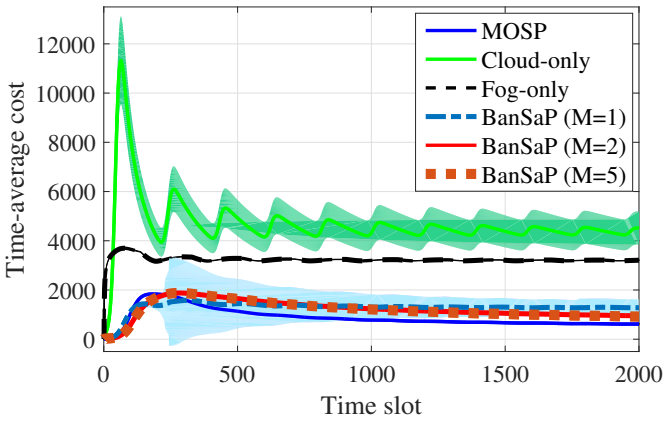

Fig. 6: Comparison of average costs. The shaded region represents the cost distribution of each scheme within one standard deviation of the mean.

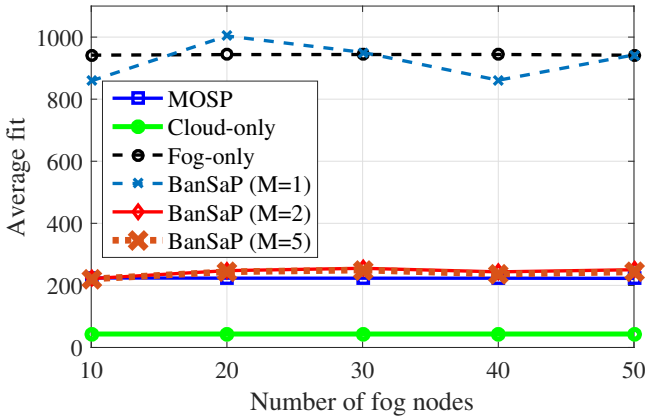

Fig. 7: Impact of network size on dynamic fit per fog node.

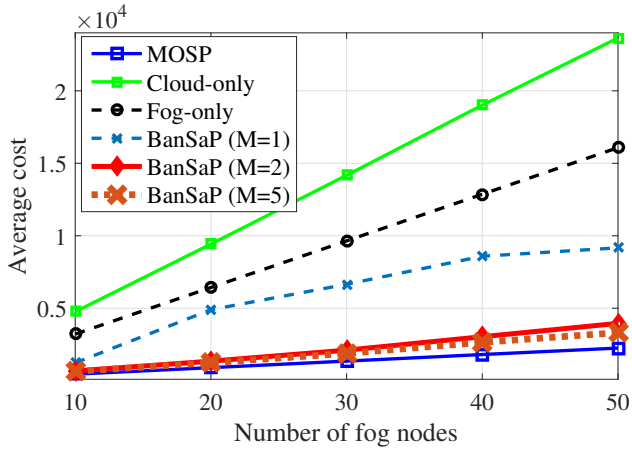

Fig. 8: Impact of network size on average network cost.

performance in the subsequent tests, coordinate sampling is adopted by BanSaP with $M = 1$, while uniform sampling is used in BanSaP with $M \geq 2$.

Optimality and feasibility. With optimized sampling schemes for BanSaP solvers, the dynamic fit and average cost are then compared among three BanSaP variants, MOSP, and two heuristic schemes in Figs. 5 and 6. Without queueing at the fog side, the cloud-only scheme has much lower dynamic fit since all user demands are offloaded to the remote cloud. However, it incurs a much higher average cost (service latency) as the network latency between fog and cloud becomes high due to the large offloading amount. By increasing the amount of feedback, the BanSaP solver tends to have a lower fit and a lower average cost, both of which are comparable to those of MOSP when $M \geq 2$. On the other hand, the BanSaP with only one-point bandit feedback still has a similar fit relative to the fog-only scheme, but enjoys much lower cost. Interestingly enough, when the variance (cf. the shaded area in Fig. 6) of the one-point BanSaP's cost is high, it markedly varnishes when multiple function values become available, which corroborates our claims in Theorems 1-2.

Effect of network size. The third test evaluates the performance of all schemes under different number of fog nodes (i.e., network size). For each algorithm, the fit averaged over all fog nodes and time is presented in Fig. 7, and the cost averaged over the time is shown in Fig. 8. Clearly, the one-point BanSaP has lower average fit than the fog-only approach in most scenarios, and also incurs less average cost in all tested settings. Similar to those in Figs. 5 and 6, the average fit of BanSaP with multiple function evaluations is still comparable to that of the full-information MOSP as the network size grows. An interesting observation here is that as the number of fog nodes increases, the performance gain of the BanSaP solver with a large M becomes more evident; see e.g., Fig. 8. This implies that for a larger network, BanSaP benefits from more bandit information to learn and track the network dynamics.

VI. Conclusions and the Road Ahead

Bandit convex optimization (BCO) in dynamic environments was studied in this paper. Different from existing works in bandit settings, the focus was on a broader setting where part of the constraints are revealed after taking actions, and are also tolerable to instantaneous violations but have to be satisfied on average. The novel BCO setting fits well the emerging fog computing tasks in IoT. A class of online bandit saddle-point (BanSaP) approaches were proposed, and their online performance was rigorously analyzed. It was shown that the resultant regret bounds match those attained in BCO setups without long-term constraints. Furthermore, the BanSaP solvers can simultaneously yield sub-linear dynamic regret and fit, if the dynamic solutions vary slowly over time.

Our algorithmic and theoretical results serve as an exciting first step to *innovate online bandit learning tailored for dynamic network management tasks*, emerging from contemporary IoT applications. Interesting future directions include designing asynchronous variants of BanSaP, and incorporating predictable dynamic models in online network optimization.

ACKNOWLEDGEMENT

The authors would like to thank Yanning Shen and Qing Ling for helpful feedback on the early version of this manuscript.

REFERENCES

[1] F. Samie, V. Tsoutsouras, S. Xydis, L. Bauer, D. Soudris, and J. Henkel, "Distributed QoS management for Internet of Things under resource constraints," in *Proc. Intl. Conf. on Hardware/Software Codesign and System Synthesis*, Pittsburgh, PA, Oct. 2016, pp. 1–10.

[2] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, 2016.

[3] G. Lee, W. Saad, and M. Bennis, "An online secretary framework for fog network formation with minimal latency," *arXiv:1702.05569*, Apr. 2017.

[4] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, D. Soudris, and J. Henkel, "Computation offloading and resource allocation for low-power IoT edge devices," in *Proc. World Forum Internet Things*, Dec. 2016, pp. 7–12.

[5] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Comm. Surveys & Tutorials*, 2017, to appear.

[6] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint offloading and computing optimization in wireless powered mobile-edge computing systems," *IEEE Trans. Wireless Commun.*, Feb. 2017, submitted. [Online]. Available: https://arxiv.org/abs/1702.00606

[7] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile edge computing: Survey and research outlook," *arXiv preprint:1701.01090*, Jan. 2017.

[8] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. Intl. Conf. on Machine Learning*, Washington D.C., Aug. 2003.

[9] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2-3, pp. 169–192, Dec. 2007.

[10] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, Jul. 2011.

[11] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online optimization: Competing with dynamic comparators," in *Intl. Conf. on Artificial Intelligence and Statistics*, San Diego, CA, May 2015.

[12] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 647–662, Jun. 2015.

[13] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Trans. Signal Processing*, Jan. 2017 (revised), Available: https://arxiv.org/abs/1701.03974.

[14] T. Chen, Y. Shen, Q. Ling, and G. B. Giannakis, "Online learning for "thing-adaptive" fog computing in IoT," in *Proc. of Asilomar Conf.*, Pacific Grove, CA, Oct. 2017. [Online]. Available: www.dropbox.com/s/z4qnog6x0gzd2ko/TAOSP.pdf?dl=0

[15] M. J. Neely and H. Yu, "Online convex optimization with time-varying constraints," *arXiv preprint:1702.04783*, Feb. 2017.

[16] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: gradient descent without a gradient," in *Proc. of ACM SODA*, Vancouver, Canada, Jan. 2005, pp. 385–394.

[17] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback." in *Proc. Annual Conf. on Learning Theory*, Haifa, Israel, 2010, pp. 28–40.

[18] O. Shamir, "An optimal algorithm for bandit and zero-order convex optimization with two-point feedback," *Journal of Machine Learning Research*, vol. 18, no. 52, pp. 1–11, 2017.

[19] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. and Trends in Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

[20] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: The power of two function evaluations," *IEEE Trans. Inform. Theory*, vol. 61, no. 5, pp. 2788–2806, May 2015.

[21] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, Apr. 2017.

[22] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: Online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 13, pp. 2503–2528, Sep 2012.

[23] B. Awerbuch and R. D. Kleinberg, "Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches," in *Proc. ACM Symp. on Theory of Computing*, Chicago, IL, Jun. 2004, pp. 45–53.

[24] Y.-H. Kao, K. Wright, B. Krishnamachari, and F. Bai, "Online learning for wireless distributed computing," *arXiv preprint:1611.02830*, Nov. 2016. [Online]. Available: https://arxiv.org/abs/1611.02830

[25] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[26] T. Chen, A. Mokhtari, X. Wang, A. Ribeiro, and G. B. Giannakis, "Stochastic averaging for constrained optimization with application to online resource allocation," *IEEE Trans. Signal Processing*, vol. 65, no. 12, pp. 3078–3093, Jun. 2017.

[27] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Info. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.

[28] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Networking*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[29] H. Huang, Q. Ling, W. Shi, and J. Wang, "Collaborative resource allocation over a hybrid cloud center and edge server network," *Journal of Computational Mathematics*, 2017, to appear.

[30] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Operations Research*, vol. 63, no. 5, pp. 1227–1244, Sep. 2015.

[31] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena scientific, 1999.

APPENDIX

The proof generalizes the result in [15] from static regret with full-information gradient feedback to the dynamic regret with partial-information bandit feedback.

A. Supporting lemmas

Before proving the main theory, we first establish several key lemmas and propositions. The following lemma establishes the unbiasedness of one- and two-point estimations [16], [17].

Lemma 1. *With* $\mathbf{u}$ *drawn uniformly from the surface of the unit ball* $\mathbb{S} := \{\mathbf{u} : \|\mathbf{u}\| = 1\} \subseteq \mathbb{R}^d$*, we have for given a constant* $\delta > 0$ *that*

$$\mathbb{E}_{\mathbf{u}}\left[\frac{d}{\delta} f_t(\mathbf{x}+\delta\mathbf{u})\mathbf{u}\right] = \nabla \check{f}_t(\mathbf{x}) \tag{35}$$

where $\nabla \check{f}_t(\mathbf{x})$ *is the gradient of the smoothed function* $\check{f}_t(\mathbf{x}) := \mathbb{E}_{\mathbf{v}}[f_t(\mathbf{x}+\delta\mathbf{v})]$ *with* $\mathbf{v}$ *drawn from a unit ball* $\mathbb{B}$*, and* d *is the dimension of the variable* $\mathbf{x}$*. Likewise, for the two-point case, we have that*

$$\mathbb{E}_{\mathbf{u}}\left[\frac{d}{2\delta}\Big(f_t(\mathbf{x}+\delta\mathbf{u}) - f_t(\mathbf{x}-\delta\mathbf{u})\Big)\mathbf{u}\right] = \nabla \check{f}_t(\mathbf{x}). \tag{36}$$

Lemma 1 provides valuable insights for performing gradient-based algorithms in bandit setting. Namely, $\hat{\nabla}^1 f_t(\hat{\mathbf{x}}_t)$ and $\hat{\nabla}^2 f_t(\hat{\mathbf{x}}_t)$ are the unbiased gradient estimators of the *smoothed* function $\check{f}_t(\mathbf{x})$, which is an approximation of $f_t(\mathbf{x})$. Note that (as1)-(as2) also imply that the smoothed function $\check{f}_t(\mathbf{x})$ is convex and G-Lipschitz continuous [17], which will be used frequently in the subsequent analysis.

The following lemma establishes the norm (or variance) of one- and two-point gradient estimations [16], [17].

Lemma 2. *For the gradient* $\hat{\nabla}^1 f_t(\hat{\mathbf{x}}_t)$ *in* (12)*, we have that*

$$\|\hat{\nabla}^1 f_t(\hat{\mathbf{x}}_t)\| \leq \frac{d}{\delta} F \tag{37}$$

where F is an upper-bound of the function. For the gradient estimator $\hat{\nabla}^2 f_t(\hat{\mathbf{x}}_t)$ in (15)*, we have that*

$$\|\hat{\nabla}^2 f_t(\hat{\mathbf{x}}_t)\| \leq dG \tag{38}$$

where G is the Lipschitz constant of the loss function.

Proof: For the gradient $\hat{\nabla}^1 f_t(\hat{\mathbf{x}}_t)$ in (12), it holds that

$$\|\hat{\nabla}^1 f_t(\hat{\mathbf{x}}_t)\| = \frac{d}{\delta}|f_t(\hat{\mathbf{x}}_t + \delta\mathbf{u}_t)|\|\mathbf{u}_t\| \leq \frac{d}{\delta}F \tag{39}$$

where F is an upper-bound of the function. Likewise for $\hat{\nabla}^2 f_t(\hat{\mathbf{x}}_t)$ in (15), we have that

$$\begin{aligned}\|\hat{\nabla}^2 f_t(\hat{\mathbf{x}}_t)\| &= \left\| \frac{d}{2\delta}\big(f_t(\hat{\mathbf{x}}_t + \delta\mathbf{u}_t) - f_t(\hat{\mathbf{x}}_t - \delta\mathbf{u}_t)\big)\mathbf{u}_t \right\| \\ &= \frac{d}{2\delta}|f_t(\hat{\mathbf{x}}_t + \delta\mathbf{u}_t) - f_t(\hat{\mathbf{x}}_t - \delta\mathbf{u}_t)|\|\mathbf{u}_t\| \\ &\leq \frac{dG}{2\delta}\|2\delta\mathbf{u}_t\| \leq dG \end{aligned} \tag{40}$$

where G is the Lipschitz constant of the loss function and $\|\mathbf{u}_t\| = 1$ by design. Thus, the proof is complete. ∎

Having bounded the norm of stochastic gradients, the next lemma is useful to ensure feasibility of actual online actions.

Lemma 3 ([16, Observation 2])**.** *Consider a constant $r > 0$ so that $r\mathbb{B} \subseteq \mathcal{X}$, where $\mathbb{B} := \{\mathbf{v} : \|\mathbf{v}\| \leq 1\} \subseteq \mathbb{R}^d$ is the unit ball. If we choose $\gamma = \delta/r$, and the iterate satisfies $\hat{\mathbf{x}}_t \in (1-\gamma)\mathcal{X}$, then $\hat{\mathbf{x}}_t + \delta\mathbf{u}_t \in \mathcal{X}$, where $\mathbf{u}_t$ is drawn uniformly from the unit sphere $\mathbb{S} := \{\mathbf{u} : \|\mathbf{u}\| = 1\} \subseteq \mathbb{R}^d$.*

The next lemma is crucial to establish the dynamic fit [15].

Lemma 4. *Considering the BanSaP recursion* (13)*, we have the following bound for the cumulative constraint violation*

$$\sum_{t=1}^{T} \mathbf{g}_t(\hat{\mathbf{x}}_t) \leq \frac{\boldsymbol{\lambda}_{T+1}}{\mu} + \frac{G^2 T\mathbf{1}}{2\beta} + \frac{\beta}{2}\sum_{t=1}^{T}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2\mathbf{1} \tag{41}$$

where $\mu > 0$ is the stepsize of the dual iteration (8)*, and $\beta > 0$ is a pre-defined constant.*

Proof: From the n-th entry of $\boldsymbol{\lambda}$ in (13), we have

$$\begin{aligned}\lambda_{t+1}^n &\geq \lambda_t^n + \mu(g_t^n(\hat{\mathbf{x}}_t) + \nabla^\top g_t^n(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t)) \\ &\overset{(a)}{\geq} \lambda_t^n + \mu g_t^n(\hat{\mathbf{x}}_t) - \frac{\mu}{2\beta}\|\nabla g_t^n(\hat{\mathbf{x}}_t)\|^2 - \frac{\mu\beta}{2}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2 \\ &\overset{(b)}{\geq} \lambda_t^n + \mu g_t^n(\hat{\mathbf{x}}_t) - \frac{\mu G^2}{2\beta} - \frac{\mu\beta}{2}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2 \end{aligned} \tag{42}$$

where (a) uses the Cauchy-Schwarz inequality, and (b) follows from the bound on the gradients in (as2). The proof is then complete after summing up (42) over $t = 1, \ldots, T$. ∎

Lemma 5. *Consider the BanSaP recursions* (11) *and* (13) *with a generic gradient $\hat{\nabla} f_t(\hat{\mathbf{x}}_t)$, which is estimated from one- or multi-point feedback. The following holds $\forall \mathbf{x} \in (1-\gamma)\mathcal{X}$*

$$\begin{aligned}&\frac{1}{\mu}\mathbb{E}[\Delta(\boldsymbol{\lambda}_t)] \leq \check{f}_t(\mathbf{x}) - \mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t)] + \mathbb{E}[\boldsymbol{\lambda}_t^\top \mathbf{g}_t(\mathbf{x})] + 2\mu G^2 R^2 \\ &+ \frac{1}{2\alpha}\mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}_t\|^2] - \frac{1}{2\alpha}\mathbb{E}[\|\mathbf{x} - \hat{\mathbf{x}}_{t+1}\|^2] + \alpha\|\hat{\nabla} f_t(\hat{\mathbf{x}}_t)\|^2 \end{aligned} \tag{43}$$

where the constants G, R and F are as in (as2) and (as3).

Proof: Taking the norm square in (13), we have

$$\begin{aligned}\|\boldsymbol{\lambda}_{t+1}\|^2 &\leq \|\boldsymbol{\lambda}_t\|^2 + 2\mu\boldsymbol{\lambda}_t^\top(\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t)) \\ &+ \mu^2\|\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t)\|^2 \\ &\leq \|\boldsymbol{\lambda}_t\|^2 + 2\mu\boldsymbol{\lambda}_t^\top(\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t)) \\ &+ 2\mu^2\|\mathbf{g}_t(\hat{\mathbf{x}}_t)\|^2 + 2\mu^2\|\nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t)\|^2. \end{aligned} \tag{44}$$

With $\Delta(\boldsymbol{\lambda}_t) := \frac{1}{2}(\|\boldsymbol{\lambda}_{t+1}\|^2 - \|\boldsymbol{\lambda}_t\|^2)$, (44) implies that

$$\frac{1}{\mu}\Delta(\boldsymbol{\lambda}_t) \leq \boldsymbol{\lambda}_t^\top(\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t)) + 2\mu G^2 R^2 \tag{45}$$

where G and R are the bounds on the gradient and the radius of the feasible set.

On the other hand, recall that the primal iterate $\hat{\mathbf{x}}_{t+1}$ is the optimal solution to the following optimization problem

$$\hat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x} \in (1-\gamma)\mathcal{X}} \hat{\nabla}_{\mathbf{x}}^\top \mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda}_t)(\mathbf{x} - \hat{\mathbf{x}}_t) + \frac{1}{2\alpha}\|\mathbf{x} - \hat{\mathbf{x}}_t\|^2. \tag{46}$$

Recalling the definition of $\hat{\nabla}_{\mathbf{x}}\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda}_t)$, we thus have that

$$\begin{aligned}\hat{\mathbf{x}}_{t+1} = \arg\min_{\mathbf{x} \in (1-\gamma)\mathcal{X}} &\hat{\nabla}^\top f_t(\hat{\mathbf{x}}_t)(\mathbf{x} - \hat{\mathbf{x}}_t) \\ &+ \boldsymbol{\lambda}_t^\top(\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\mathbf{x} - \hat{\mathbf{x}}_t)) + \frac{1}{2\alpha}\|\mathbf{x} - \hat{\mathbf{x}}_t\|^2 \end{aligned} \tag{47}$$

where we add $\boldsymbol{\lambda}_t^\top \mathbf{g}_t(\mathbf{x}_t)$ to the RHS of (46). Note that it will not change the minimizer of (46), since the added term is constant, and not coupled with the variable $\mathbf{x}$.

To connect (45) with the bound obtained in (47), adding $\hat{\nabla}^\top f_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t) + \frac{1}{2\alpha}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2$ to the RHS of (45), we have that

$$\begin{aligned}&\frac{1}{\mu}\Delta(\boldsymbol{\lambda}_t) + \hat{\nabla}^\top f_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t) + \frac{1}{2\alpha}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2 \\ &\leq \boldsymbol{\lambda}_t^\top\big(\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t)\big) + \frac{1}{2\alpha}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2 \\ &\quad + \hat{\nabla}^\top f_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t) + 2\mu G^2 R^2. \end{aligned} \tag{48}$$

Note that $\hat{\mathbf{x}}_{t+1}$ is the minimizer of (47), where the objective on the RHS of (47) is strongly-convex, thus we have that

$$\begin{aligned}&\frac{1}{\mu}\Delta(\boldsymbol{\lambda}_t) + \hat{\nabla}^\top f_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t) + \frac{1}{2\alpha}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2 \\ &\leq \boldsymbol{\lambda}_t^\top\big(\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\mathbf{x} - \hat{\mathbf{x}}_t)\big) + \frac{1}{2\alpha}\|\mathbf{x} - \hat{\mathbf{x}}_t\|^2 + 2\mu G^2 R^2 \\ &\quad + \hat{\nabla}^\top f_t(\hat{\mathbf{x}}_t)(\mathbf{x} - \hat{\mathbf{x}}_t) - \frac{1}{2\alpha}\|\mathbf{x} - \hat{\mathbf{x}}_{t+1}\|^2 \\ &\overset{(a)}{\leq} \boldsymbol{\lambda}_t^\top \mathbf{g}_t(\mathbf{x}) + \hat{\nabla}^\top f_t(\hat{\mathbf{x}}_t)(\mathbf{x} - \hat{\mathbf{x}}_t) + 2\mu G^2 R^2 \\ &\quad + \frac{1}{2\alpha}\|\mathbf{x} - \hat{\mathbf{x}}_t\|^2 - \frac{1}{2\alpha}\|\mathbf{x} - \hat{\mathbf{x}}_{t+1}\|^2, \ \forall \mathbf{x} \in (1-\gamma)\mathcal{X} \end{aligned} \tag{49}$$

where (a) uses the non-negativity that $\boldsymbol{\lambda}_t \geq \mathbf{0}$, and the convexity such that $\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\mathbf{x} - \hat{\mathbf{x}}_t) \leq \mathbf{g}_t(\mathbf{x})$.

Using the Cauchy-Schwarz inequality, we have that

$$-\hat{\nabla}^\top f_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t) \leq \alpha\|\hat{\nabla} f_t(\hat{\mathbf{x}}_t)\|^2 + \frac{\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2}{4\alpha}. \tag{50}$$

Plugging (50) into (49) and rearranging terms, for $\forall \mathbf{x} \in (1-\gamma)\mathcal{X}$, we have that

$$\begin{aligned}&\frac{1}{\mu}\Delta(\boldsymbol{\lambda}_t) + \frac{1}{4\alpha}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2 \\ &\leq \boldsymbol{\lambda}_t^\top \mathbf{g}_t(\mathbf{x}) + \hat{\nabla}^\top f_t(\hat{\mathbf{x}}_t)(\mathbf{x} - \hat{\mathbf{x}}_t) + 2\mu G^2 R^2 \\ &\quad + \frac{1}{2\alpha}\|\mathbf{x} - \hat{\mathbf{x}}_t\|^2 - \frac{1}{2\alpha}\|\mathbf{x} - \hat{\mathbf{x}}_{t+1}\|^2 + \alpha\|\hat{\nabla} f_t(\hat{\mathbf{x}}_t)\|^2. \end{aligned} \tag{51}$$

Taking expectation over $\mathbf{u}_t$ on both side of (51) conditioning on $\hat{\mathbf{x}}_t$, it follows that

$$\frac{1}{\mu}\mathbb{E}[\Delta(\boldsymbol{\lambda}_t)] + \frac{1}{4\alpha}\mathbb{E}[\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2]$$

$$\leq \boldsymbol{\lambda}_t^\top \mathbf{g}_t(\mathbf{x}) + \mathbb{E}\left[\hat{\nabla}^\top f_t(\hat{\mathbf{x}}_t)\,(\mathbf{x}-\hat{\mathbf{x}}_t)\right] + 2\mu G^2 R^2$$
$$+ \frac{1}{2\alpha}\|\mathbf{x}-\hat{\mathbf{x}}_t\|^2 - \frac{1}{2\alpha}\mathbb{E}[\|\mathbf{x}-\hat{\mathbf{x}}_{t+1}\|^2] + \alpha\|\hat{\nabla} f_t(\hat{\mathbf{x}}_t)\|^2$$
$$\overset{(c)}{=} \boldsymbol{\lambda}_t^\top \mathbf{g}_t(\mathbf{x}) + \nabla^\top \check{f}_t(\hat{\mathbf{x}}_t)\,(\mathbf{x}-\hat{\mathbf{x}}_t) + 2\mu G^2 R^2$$
$$+ \frac{1}{2\alpha}\|\mathbf{x}-\hat{\mathbf{x}}_t\|^2 - \frac{1}{2\alpha}\mathbb{E}[\|\mathbf{x}-\hat{\mathbf{x}}_{t+1}\|^2] + \alpha\|\hat{\nabla} f_t(\hat{\mathbf{x}}_t)\|^2 \quad (52)$$

where (c) holds since the randomness $\mathbf{u}_t$ in $\hat{\nabla} f_t(\hat{\mathbf{x}}_t)$ is independent of $\hat{\mathbf{x}}_t$, and $\hat{\nabla} f_t(\hat{\mathbf{x}}_t)$ is an unbiased estimator of $\nabla \check{f}_t(\hat{\mathbf{x}}_t)$.

The convexity of $f_t(\mathbf{x})$ implies that $\check{f}_t(\mathbf{x})$ is also convex, and thus $\nabla^\top \check{f}_t(\hat{\mathbf{x}}_t)\,(\mathbf{x}-\hat{\mathbf{x}}_t) \leq \check{f}_t(\mathbf{x}) - \check{f}_t(\hat{\mathbf{x}}_t)$. Plugging into (52) and taking expectation over all possible $\hat{\mathbf{x}}_t$, it follows that

$$\frac{1}{\mu}\mathbb{E}[\Delta(\boldsymbol{\lambda}_t)] + \frac{1}{4\alpha}\mathbb{E}[\|\hat{\mathbf{x}}_{t+1}-\hat{\mathbf{x}}_t\|^2] \quad (53)$$
$$\leq \check{f}_t(\mathbf{x}) - \mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t)] + \mathbb{E}[\boldsymbol{\lambda}_t^\top \mathbf{g}_t(\mathbf{x})] + 2\mu G^2 R^2$$
$$+ \frac{1}{2\alpha}\mathbb{E}[\|\mathbf{x}-\hat{\mathbf{x}}_t\|^2] - \frac{1}{2\alpha}\mathbb{E}[\|\mathbf{x}-\hat{\mathbf{x}}_{t+1}\|^2] + \alpha\mathbb{E}[\|\hat{\nabla} f_t(\hat{\mathbf{x}}_t)\|^2]$$

which completes the proof by dropping the nonnegative term $\mathbb{E}[\|\hat{\mathbf{x}}_{t+1}-\hat{\mathbf{x}}_t\|^2]$ in the LHS. ■

B. Proof of Theorem 1

With $\gamma = \delta/r$, the feasibility of actions $\{\mathbf{x}_{1,t}\}$ readily follows from Lemma 3, i.e., $\mathbf{x}_{1,t} \in \mathcal{X}$, $\forall t$. To prove the dynamic regret and fit bounds, the following result is needed.

Lemma 6. *For the BanSaP recursions* (11)-(13)*, if we choose* $\alpha = \mu = \mathcal{O}(T^{-\frac{3}{4}})$ *and* $\delta = \mathcal{O}(T^{-\frac{1}{4}})$*, the dual iterates are uniformly bounded by* $\|\boldsymbol{\lambda}_t\| \leq C = \mathcal{O}(1)$*, with the constant* C *given by*

$$C := \max\left\{2GR, \left(\frac{1}{\eta}+1\right)GR + \frac{2G^2R^2\mu}{\eta} + \frac{d^2F^2\alpha}{\eta\delta^2} + \frac{\mu R^2}{2\alpha\eta}\right\} \quad (54)$$

where the constants G*,* R*, and* η *are as in (as2)-(as4).*

Proof: Plugging the bounded norm of the one-point gradient estimator (37) into (43), it holds that

$$\frac{1}{\mu}\mathbb{E}[\Delta(\boldsymbol{\lambda}_t)] \leq GR + \mathbb{E}[\boldsymbol{\lambda}_t^\top \mathbf{g}_t(\mathbf{x})] + 2\mu G^2R^2 + \frac{d^2F^2\alpha}{\delta^2}$$
$$+ \frac{1}{2\alpha}\mathbb{E}[\|\mathbf{x}-\hat{\mathbf{x}}_t\|^2] - \frac{1}{2\alpha}\mathbb{E}[\|\mathbf{x}-\hat{\mathbf{x}}_{t+1}\|^2] \quad (55)$$

where we used the Lipschitz condition on (43); i.e.,

$$\mathbb{E}[\check{f}_t(\mathbf{x}) - \check{f}_t(\hat{\mathbf{x}}_t)] \leq GR. \quad (56)$$

Selecting the interior point $\mathbf{x} = \tilde{\mathbf{x}} \in (1-\gamma)\mathcal{X}$ so that $\mathbf{g}_t(\tilde{\mathbf{x}}) \leq -\eta\mathbf{1}$, it follows from (55) that

$$\frac{1}{\mu}\mathbb{E}[\Delta(\boldsymbol{\lambda}_t)] \leq GR - \eta\mathbb{E}[\boldsymbol{\lambda}_t^\top \mathbf{1}] + 2\mu G^2R^2 + \frac{d^2F^2\alpha}{\delta^2}$$
$$+ \frac{1}{2\alpha}\mathbb{E}[\|\tilde{\mathbf{x}}-\hat{\mathbf{x}}_t\|^2] - \frac{1}{2\alpha}\mathbb{E}[\|\tilde{\mathbf{x}}-\hat{\mathbf{x}}_{t+1}\|^2]. \quad (57)$$

Using $-\eta\boldsymbol{\lambda}_t^\top\mathbf{1} = -\eta\|\boldsymbol{\lambda}_t\|_1 \leq -\eta\|\boldsymbol{\lambda}_t\|$, we arrive at

$$\frac{1}{\mu}\mathbb{E}[\Delta(\boldsymbol{\lambda}_t)] \leq GR - \eta\mathbb{E}[\|\boldsymbol{\lambda}_t\|] + 2\mu G^2R^2 + \frac{d^2F^2\alpha}{\delta^2}$$
$$+ \frac{1}{2\alpha}\mathbb{E}[\|\tilde{\mathbf{x}}-\hat{\mathbf{x}}_t\|^2] - \frac{1}{2\alpha}\mathbb{E}[\|\tilde{\mathbf{x}}-\hat{\mathbf{x}}_{t+1}\|^2]. \quad (58)$$

Now we are ready to show that the norm of the dual variable is uniformly bounded by a constant C that is independent of time; that is, $\|\boldsymbol{\lambda}_t\| \leq C$, $\forall t$.

For $1 \leq t \leq \frac{1}{\mu}$, it follows readily that

$$\|\boldsymbol{\lambda}_t\| \leq \|\boldsymbol{\lambda}_{t-1}\| + \mu\|\mathbf{g}_t(\hat{\mathbf{x}}_t) + \nabla^\top \mathbf{g}_t(\hat{\mathbf{x}}_t)(\hat{\mathbf{x}}_{t+1}-\hat{\mathbf{x}}_t)\|$$
$$\leq \|\boldsymbol{\lambda}_{t-1}\| + 2\mu GR \leq \|\boldsymbol{\lambda}_1\| + 2\mu t GR \leq C \quad (59)$$

where the last inequality follows from $\boldsymbol{\lambda}_1 = \mathbf{0}$, $t \leq 1/\mu$, and the definition of C in (54).

For $\frac{1}{\mu} \leq t \leq T$, we will prove the claim by contradiction. Assume T_0 is the first slot for which $\|\boldsymbol{\lambda}_{T_0}\| > C$. Therefore, we have $\|\boldsymbol{\lambda}_{T_0}\| > C \geq \|\boldsymbol{\lambda}_{T_0-\frac{1}{\mu}}\|$, which after recalling (58) and the definition of $\Delta(\boldsymbol{\lambda}_t)$, yields

$$\frac{1}{\mu}\sum_{t=T_0-\frac{1}{\mu}}^{T_0-1}\mathbb{E}[\Delta(\boldsymbol{\lambda}_t)] = \frac{1}{2\mu}\left(\mathbb{E}\left[\|\boldsymbol{\lambda}_{T_0}\|^2 - \|\boldsymbol{\lambda}_{T_0-\frac{1}{\mu}}\|^2\right]\right) > 0. \quad (60)$$

On the other hand however, summing up (58), we obtain

$$\frac{1}{\mu}\sum_{t=T_0-\frac{1}{\mu}}^{T_0-1}\mathbb{E}[\Delta(\boldsymbol{\lambda}_t)] \leq \frac{GR}{\mu} - \eta\sum_{t=T_0-\frac{1}{\mu}}^{T_0-1}\mathbb{E}[\|\boldsymbol{\lambda}_t\|] + 2G^2R^2$$
$$+ \frac{d^2F^2\alpha}{\mu\delta^2} + \frac{1}{2\alpha}\mathbb{E}[\|\tilde{\mathbf{x}}-\hat{\mathbf{x}}_{T_0-\frac{1}{\mu}}\|^2] - \frac{1}{2\alpha}\mathbb{E}[\|\tilde{\mathbf{x}}-\hat{\mathbf{x}}_{T_0}\|^2]$$
$$\overset{(a)}{\leq} \frac{GR}{\mu} - \eta\sum_{t=T_0-\frac{1}{\mu}}^{T_0-1}\mathbb{E}[\|\boldsymbol{\lambda}_t\|] + 2G^2R^2 + \frac{d^2F^2\alpha}{\mu\delta^2} + \frac{R^2}{2\alpha} \quad (61)$$

where (a) uses again the bound $\|\tilde{\mathbf{x}}-\hat{\mathbf{x}}_{T_0-\frac{1}{\mu}}\| \leq R$.

Note that since $\|\boldsymbol{\lambda}_{T_0}\| > C$ and $\|\boldsymbol{\lambda}_{T_0}\| - \|\boldsymbol{\lambda}_{T_0-1}\| \leq 2\mu GR$, we have that

$$\|\boldsymbol{\lambda}_{T_0-\tau}\| > C - 2\tau\mu GR. \quad (62)$$

Combining (61) with (62), we deduce

$$\frac{1}{\mu}\sum_{t=T_0-\frac{1}{\mu}}^{T_0-1}\mathbb{E}[\Delta(\boldsymbol{\lambda}_t)] \leq \frac{GR}{\mu} - \frac{C\eta}{\mu} + \frac{\eta GR}{\mu} + 2G^2R^2 + \frac{d^2F^2\alpha}{\mu\delta^2} + \frac{R^2}{2\alpha}. \quad (63)$$

Together with (60), recursion (63) implies that

$$C < \frac{GR}{\eta} + GR + \frac{2G^2R^2\mu}{\eta} + \frac{d^2F^2\alpha}{\eta\delta^2} + \frac{\mu R^2}{2\alpha\eta} \quad (64)$$

which contradicts the definition of C in (54). Hence, there is no T_0 satisfying $\|\boldsymbol{\lambda}_t\| \leq C$, which implies that $\|\boldsymbol{\lambda}_t\| \leq C$, $\forall t$.

By choosing the stepsizes $\alpha = \mu = \mathcal{O}(T^{-\frac{3}{4}})$, and the parameter $\delta = \mathcal{O}(T^{-\frac{1}{4}})$, it follows that

$$C = \mathcal{O}\left(\frac{GR}{\eta} + GR + \frac{2G^2R^2}{\eta T^{\frac{3}{4}}} + \frac{d^2F^2}{\eta T^{\frac{1}{4}}} + \frac{R^2}{2\eta}\right) = \mathcal{O}(1) \quad (65)$$

which completes the proof of the lemma. ■

Dynamic regret in Theorem 1: Recall that $\mathbf{x}_t^*$ is the minimizer of the following time-varying problem (20), and note that $(1-\gamma)\mathbf{x}_t^* \in (1-\gamma)\mathcal{X}$. Hence, plugging $(1-\gamma)\mathbf{x}_t^*$ into (43), we have

$$\frac{1}{\mu}\mathbb{E}[\Delta(\boldsymbol{\lambda}_t)] \leq \check{f}_t((1-\gamma)\mathbf{x}_t^*) - \mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t)]$$
$$+ \frac{1}{2\alpha}\mathbb{E}[\|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_t\|^2] - \frac{1}{2\alpha}\mathbb{E}[\|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_{t+1}\|^2]$$
$$+ \mathbb{E}[\boldsymbol{\lambda}_t^\top \mathbf{g}_t((1-\gamma)\mathbf{x}_t^*)] + \frac{\alpha}{\delta^2}d^2F^2 + 2\mu G^2R^2. \quad (66)$$

From the Lipschitz condition, we can bound the inner

product in (66) by

$$\begin{aligned}&\mathbb{E}[\boldsymbol{\lambda}_t^\top \mathbf{g}_t((1-\gamma)\mathbf{x}_t^*)]\\ \leq&\mathbb{E}[\boldsymbol{\lambda}_t^\top(\mathbf{g}_t(\mathbf{x}_t^*)+\gamma GR\cdot\mathbf{1})] \overset{(a)}{\leq} \gamma GR\mathbb{E}[\|\boldsymbol{\lambda}_t\|] \overset{(b)}{\leq} \gamma GR\|\bar{\boldsymbol{\lambda}}\| \quad (67)\end{aligned}$$

where (a) follows from $\boldsymbol{\lambda}_t^\top \mathbf{g}_t(\mathbf{x}_t^*) \leq 0$ since $\mathbf{g}_t(\mathbf{x}_t^*) \leq \mathbf{0}$, and $\boldsymbol{\lambda}_t \geq \mathbf{0}$; and (b) uses the upper bound of $\|\bar{\boldsymbol{\lambda}}\| := \max_t \|\boldsymbol{\lambda}_t\|$. The two distance terms in (66) can be bounded by

$$\begin{aligned}&\|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_t\|^2 - \|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_{t+1}\|^2\\ =&\|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_t\|^2 - \|(1-\gamma)\mathbf{x}_{t-1}^* - \hat{\mathbf{x}}_t\|^2\\ &+\|(1-\gamma)\mathbf{x}_{t-1}^* - \hat{\mathbf{x}}_t\|^2 - \|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_{t+1}\|^2\\ =&(1-\gamma)\|\mathbf{x}_t^* - \mathbf{x}_{t-1}^*\|\|(1-\gamma)(\mathbf{x}_t^* + \mathbf{x}_{t-1}^*) - 2\hat{\mathbf{x}}_t\|\\ &+\|(1-\gamma)\mathbf{x}_{t-1}^* - \hat{\mathbf{x}}_t\|^2 - \|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_{t+1}\|^2. \quad (68)\end{aligned}$$

Using the triangle inequality, it follows that

$$\begin{aligned}&\|(1-\gamma)(\mathbf{x}_t^* + \mathbf{x}_{t-1}^*) - 2\hat{\mathbf{x}}_t\|\\ \leq&\|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_t\| + \|(1-\gamma)\mathbf{x}_{t-1}^* - \hat{\mathbf{x}}_t\| \leq 2R \quad (69)\end{aligned}$$

which together with (68), implies that

$$\begin{aligned}&\|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_t\|^2 - \|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_{t+1}\|^2\\ \leq& 2(1-\gamma)R\|\mathbf{x}_t^* - \mathbf{x}_{t-1}^*\| + \|(1-\gamma)\mathbf{x}_{t-1}^* - \hat{\mathbf{x}}_t\|^2\\ &- \|(1-\gamma)\mathbf{x}_t^* - \hat{\mathbf{x}}_{t+1}\|^2. \quad (70)\end{aligned}$$

Plugging (67) and (70) into (66), and summing up over $t = 1, \ldots, T$, we find

$$\begin{aligned}&\frac{1}{2\mu}\left(\mathbb{E}[\|\boldsymbol{\lambda}_{T+1}\|^2 - \|\boldsymbol{\lambda}_1\|^2]\right) + \sum_{t=1}^{T}\left(\mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t)] - \check{f}_t((1-\gamma)\mathbf{x}_t^*)\right)\\ \leq&\gamma GR\|\bar{\boldsymbol{\lambda}}\|T + \sum_{t=1}^{T}\frac{(1-\gamma)R}{\alpha}\|\mathbf{x}_t^* - \mathbf{x}_{t-1}^*\| + 2\mu G^2R^2T + \frac{\alpha d^2F^2T}{\delta^2}\\ &+\frac{1}{2\alpha}\left(\mathbb{E}\left[\,\|(1-\gamma)\mathbf{x}_0^* - \hat{\mathbf{x}}_1\|^2\,\right] - \mathbb{E}\left[\,\|(1-\gamma)\mathbf{x}_T^* - \hat{\mathbf{x}}_{T+1}\|^2\,\right]\right)\\ \overset{(c)}{\leq}&\gamma GR\|\bar{\boldsymbol{\lambda}}\|T + \frac{R}{\alpha}V(\mathbf{x}_{1:T}^*) + 2\mu G^2R^2T + \frac{R^2}{2\alpha} + \frac{\alpha d^2F^2T}{\delta^2} \quad (71)\end{aligned}$$

where (c) uses $\|(1-\gamma)\mathbf{x}_0^* - \hat{\mathbf{x}}_1\| \leq \|\mathbf{x}_0^* - \hat{\mathbf{x}}_1\| \leq R$, and $\|(1-\gamma)\mathbf{x}_T^* - \hat{\mathbf{x}}_{T+1}\|^2 \geq 0$, and the accumulated variation of the per-slot minimizers defined as $V(\mathbf{x}_{1:T}^*) := \sum_{t=1}^{T}\|\mathbf{x}_t^* - \mathbf{x}_{t-1}^*\|$.

Since $\mathbb{E}[\|\boldsymbol{\lambda}_{T+1}\|^2] \geq 0$, initializing the dual variable with $\boldsymbol{\lambda}_1 = \mathbf{0}$, and rearranging (71), we have that

$$\begin{aligned}&\sum_{t=1}^{T}\left(\mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t)] - \check{f}_t((1-\gamma)\mathbf{x}_t^*)\right)\\ \leq&\gamma GR\|\bar{\boldsymbol{\lambda}}\|T + \frac{R}{\alpha}V(\mathbf{x}_{1:T}^*) + 2\mu G^2R^2T + \frac{R^2}{2\alpha} + \frac{\alpha d^2F^2T}{\delta^2}. \quad (72)\end{aligned}$$

The iterates $\{\hat{\mathbf{x}}_t\}$ in this bound are not the actual decisions taken by the learner. To obtain the regret bound, our next step is to decompose the regret as

$$\begin{aligned}\sum_{t=1}^{T}\left(\mathbb{E}[f_t(\mathbf{x}_{1,t})] - f_t(\mathbf{x}_t^*)\right) =& \sum_{t=1}^{T}\Big(\underbrace{\mathbb{E}[f_t(\mathbf{x}_{1,t})] - \mathbb{E}[\check{f}_t(\mathbf{x}_{1,t})]}_{U_1}\\ &+\underbrace{\mathbb{E}[\check{f}_t(\mathbf{x}_{1,t})] - \mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t)]}_{U_2} + \underbrace{\mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t)] - \check{f}_t((1-\gamma)\mathbf{x}_t^*))}_{U_3}\\ &+\underbrace{\check{f}_t((1-\gamma)\mathbf{x}_t^*)) - \check{f}_t(\mathbf{x}_t^*)}_{U_4} + \underbrace{\check{f}_t(\mathbf{x}_t^*) - f_t(\mathbf{x}_t^*)}_{U_5}\Big). \quad (73)\end{aligned}$$

We next bound each under-braced, starting with

$$\begin{aligned}U_1 =&\mathbb{E}\left[f_t(\mathbf{x}_{1,t}) - \mathbb{E}_{\mathbf{v}}[f_t(\mathbf{x}_{1,t} + \delta\mathbf{v}_t)]\right]\\ \overset{(d)}{\leq}&\mathbb{E}\left[f_t(\mathbf{x}_{1,t}) - f_t(\mathbb{E}_{\mathbf{v}}[\mathbf{x}_{1,t} + \delta\mathbf{v}_t])\right] \overset{(e)}{=} 0 \quad (74)\end{aligned}$$

where (d) uses Jensen's inequality, and (e) follows from $\mathbb{E}_{\mathbf{v}}[\delta\mathbf{v}_t] = \mathbf{0}$ since $\mathbf{v}_t$ is drawn from $\mathbb{B} := \{\mathbf{v} : \|\mathbf{v}\| \leq 1\}$.

Regarding the second term, it follows that

$$U_2 = \mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t + \delta\mathbf{u}_t) - \check{f}_t(\hat{\mathbf{x}}_t)] \overset{(f)}{\leq} \mathbb{E}[G\|\delta\mathbf{u}_t\|] = \delta G \quad (75)$$

where (f) uses the Lipschitz condition of $\check{f}_t(\mathbf{x})$. The third term U_3 has been already bounded as in (72).

Using the Lipschitz condition of $\check{f}_t(\mathbf{x})$, we can further bound the fourth term

$$U_4 = \check{f}_t((1-\gamma)\mathbf{x}_t^*)) - \check{f}_t(\mathbf{x}_t^*) \leq \gamma GR \quad (76)$$

and likewise for the last term for which

$$U_5 = \mathbb{E}_{\mathbf{v}}[f_t(\mathbf{x}_t^* + \delta\mathbf{v}_t)] - f_t(\mathbf{x}_t^*) \leq \mathbb{E}_{\mathbf{v}}\left[G\|\delta\mathbf{v}_t\|\right] \leq \delta G. \quad (77)$$

Plugging (72) and (74)-(77) into (73), we arrive that

$$\begin{aligned}\sum_{t=1}^{T}\left(\mathbb{E}[f_t(\mathbf{x}_{1,t})] - f_t(\mathbf{x}_t^*)\right) \leq& \frac{R}{\alpha}V(\mathbf{x}_{1:T}^*) + \frac{R^2}{2\alpha} + \frac{d^2G^2R^2\alpha T}{\delta^2}\\ &+\gamma GRT(1 + \|\bar{\boldsymbol{\lambda}}\|) + 2\mu G^2R^2T + 2G\delta T. \quad (78)\end{aligned}$$

Upon choosing $\alpha = \mu = \mathcal{O}(T^{-\frac{3}{4}})$, and $\delta = \mathcal{O}(T^{-\frac{1}{4}})$ along with $\gamma = \delta/r$, it follows that (cf. Lemma 6)

$$\mathrm{Reg}_T^{\mathrm{d}} = \mathcal{O}\Big(RV(\mathbf{x}_{1:T}^*)T^{\frac{3}{4}} + GRCT^{\frac{3}{4}} + 2G^2R^2T^{\frac{1}{4}} + d^2G^2R^2T^{\frac{3}{4}}\Big)$$

from which the proof is complete.

Dynamic fit in Theorem 1: To bound the dynamic fit, recall that the constraint violations in (41) depend on the magnitude of the dual variable and the difference of two consecutive primal iterates. The distance between iterates $\hat{\mathbf{x}}_t$ and $\hat{\mathbf{x}}_{t+1}$ can be bounded as

$$\begin{aligned}&\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\| \overset{(a)}{\leq} \left\|\alpha\hat{\nabla}_{\mathbf{x}}^1\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda}_t)\right\|\\ \overset{(b)}{\leq}&\frac{d}{\delta}|f_t(\hat{\mathbf{x}}_t + \delta\mathbf{u}_t)| + \|\nabla\mathbf{g}_t(\hat{\mathbf{x}}_t)\|\|\boldsymbol{\lambda}_t\| \overset{(c)}{\leq} \frac{\alpha dF}{\delta} + \alpha G\|\boldsymbol{\lambda}_t\| \quad (79)\end{aligned}$$

where (a) uses the non-expansive property of the projection operator, (b) relies on (12) and the Cauchy-Schwarz's inequality; and (c) uses the bounds in (as2).

On the other hand, using the Lipschitz continuity of $\mathbf{g}_t(\mathbf{x})$ and (41), it follows that

$$\begin{aligned}&\sum_{t=1}^{T}\mathbf{g}_t(\mathbf{x}_{1,t}) \leq \sum_{t=1}^{T}\mathbf{g}_t(\hat{\mathbf{x}}_t) + \delta GT\mathbf{1} \quad (80)\\ \leq&\frac{\boldsymbol{\lambda}_{T+1}}{\mu} + \frac{G^2T\mathbf{1}}{2\beta} + \frac{\beta}{2}\sum_{t=1}^{T}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2\mathbf{1} + \delta GT\mathbf{1}\\ \overset{(d)}{\leq}&\frac{\boldsymbol{\lambda}_{T+1}}{\mu} + \frac{G^2T\mathbf{1}}{2\beta} + \beta T\Big(\frac{\alpha^2d^2F^2}{\delta^2} + \alpha^2G^2\|\bar{\boldsymbol{\lambda}}\|^2\Big)\mathbf{1} + \delta GT\mathbf{1}\end{aligned}$$

where (d) uses (79), and the fact that $(a+b)^2 \leq 2(a^2+b^2)$. Taking $[\cdot]^+$ and $\|\cdot\|$ on both sides of (80), we have (cf. (21))

$$\begin{aligned}\mathrm{Fit}_T^{\mathrm{d}} \leq& \frac{\|\bar{\boldsymbol{\lambda}}\|}{\mu} + \frac{G^2\sqrt{N}T}{2\beta} + \delta G\sqrt{N}T\\ &+\beta\sqrt{N}T\left(\frac{\alpha^2d^2F^2}{\delta^2} + \alpha^2G^2\|\bar{\boldsymbol{\lambda}}\|^2\right) \quad (81)\end{aligned}$$

which establishes (24). Upon selecting $\alpha = \mathcal{O}(T^{-\frac{3}{4}})$, and $\delta =$

$\mathcal{O}(T^{-\frac{1}{4}})$, we find from Lemma 6 that $\|\bar{\boldsymbol{\lambda}}\| \leq C = \mathcal{O}(1)$. Together with $\mu = \mathcal{O}(T^{-\frac{3}{4}})$ and $\beta = \mathcal{O}(T^{\frac{1}{4}})$, it holds from (81) that

$$\begin{aligned}\mathrm{Fit}_T^{\mathrm{d}} \leq & CT^{\frac{3}{4}} + \frac{G^2\sqrt{N}T^{\frac{3}{4}}}{2} + G\sqrt{N}T^{\frac{3}{4}} \\ & + \sqrt{N}T^{\frac{5}{4}}\left(d^2F^2T^{-1} + T^{-\frac{3}{2}}G^2C^2\right) = \mathcal{O}(T^{\frac{3}{4}}) \quad (82)\end{aligned}$$

which completes the proof of (25).

C. Proof of Theorem 2

Similar to the proof of Theorem 1, feasibility of actions $\{\mathbf{x}_{1,t}, \mathbf{x}_{2,t}\}$ readily follows from Lemma 3; hence, $\mathbf{x}_{1,t}, \mathbf{x}_{2,t} \in \mathcal{X}$, $\forall t$. To prove the dynamic regret and fit bounds in this setup, the following result is needed.

Lemma 7. *For the BanSaP recursion* (11), (13)*, and* (12)*, selecting $\alpha = \mu = \mathcal{O}(T^{-\frac{1}{2}})$ ensures that the dual iterates are uniformly bounded by $\|\boldsymbol{\lambda}_t\| \leq C = \mathcal{O}(1)$, with the constant C given by*

$$C := \max\left\{2GR, \left(\frac{1}{\eta}+1\right)GR + \frac{2G^2R^2\mu}{\eta} + \frac{d^2G^2\alpha}{\eta} + \frac{\mu R^2}{2\alpha\eta}\right\} \quad (83)$$

where the constants G, R, and η are as in (as2)-(as4).

Proof: It follows steps similar to those used to prove Lemma 6. ∎

Similar to Lemma 6, Lemma 7 asserts that the dual variable in BanSaP with two-point bandit feedback is also uniformly bounded from above. Now, we are ready to prove the regret bound in Theorem 2.

Dynamic regret in Theorem 2: To obtain the regret bound in the case of two-point feedback, our first step is to connect the regret with the optimality loss induced by the sequence of iterates $\{\hat{\mathbf{x}}_t\}$, given by

$$\begin{aligned}&\frac{1}{2}\sum_{t=1}^{T}\Big(\mathbb{E}[f_t(\mathbf{x}_{1,t})] + \mathbb{E}[f_t(\mathbf{x}_{2,t})]\Big) - \sum_{t=1}^{T} f_t(\mathbf{x}_t^*) \\ &\overset{(a)}{\leq} \frac{1}{2}\sum_{t=1}^{T}\Big(\mathbb{E}[f_t(\hat{\mathbf{x}}_t)] + \delta G + \mathbb{E}[f_t(\hat{\mathbf{x}}_t)] + \delta G\Big) - \sum_{t=1}^{T} f_t(\mathbf{x}_t^*) \\ &= \sum_{t=1}^{T}\Big(\mathbb{E}[f_t(\hat{\mathbf{x}}_t)] - f_t(\mathbf{x}_t^*)\Big) + \delta G \quad (84)\end{aligned}$$

where (a) follows from the Lipschitz condition in (as2).

The LHS of (84) can be further decomposed as

$$\begin{aligned}&\sum_{t=1}^{T}\Big(\mathbb{E}[f_t(\hat{\mathbf{x}}_t)] - f_t(\mathbf{x}_t^*)\Big) \\ &= \sum_{t=1}^{T}\Big(\underbrace{\mathbb{E}[f_t(\hat{\mathbf{x}}_t)] - \mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t)]}_{U_1} + \underbrace{\mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t)] - \check{f}_t((1-\gamma)\mathbf{x}_t^*)}_{U_2} \\ &\quad + \underbrace{\check{f}_t((1-\gamma)\mathbf{x}_t^*) - \check{f}_t(\mathbf{x}_t^*)}_{U_3} + \underbrace{\check{f}_t(\mathbf{x}_t^*) - f_t(\mathbf{x}_t^*)}_{U_4}\Big). \quad (85)\end{aligned}$$

For the first term, following the steps in (74), we have that

$$\begin{aligned}U_1 \leq & \mathbb{E}\left[f_t(\hat{\mathbf{x}}_t) - \mathbb{E}_{\mathbf{v}}[f_t(\hat{\mathbf{x}}_t + \delta\mathbf{v}_t)]\right] \\ \leq & \mathbb{E}\left[f_t(\hat{\mathbf{x}}_t) - f_t(\mathbb{E}_{\mathbf{v}}[\hat{\mathbf{x}} + \delta\mathbf{v}_t])\right] \leq 0. \quad (86)\end{aligned}$$

Similar to (72), we have for the case of two-point feedback

$$\begin{aligned}\sum_{t=1}^{T} U_2 = & \sum_{t=1}^{T}\left(\mathbb{E}[\check{f}_t(\hat{\mathbf{x}}_t)] - \check{f}_t((1-\gamma)\mathbf{x}_t^*)\right) \quad (87) \\ \leq & \gamma GR\|\bar{\boldsymbol{\lambda}}\|T + \frac{R}{\alpha}V(\mathbf{x}_{1:T}^*) + 2\mu G^2R^2T + \frac{R^2}{2\alpha} + \alpha d^2G^2T.\end{aligned}$$

Using the Lipschitz condition of $\check{f}_t(\mathbf{x})$, we can bound the third term

$$U_3 = \check{f}_t((1-\gamma)\mathbf{x}_t^*)) - \check{f}_t(\mathbf{x}_t^*) \leq \gamma GR \quad (88)$$

and likewise for the last term, it follows from the Lipschitz condition of $f_t(\mathbf{x})$ that

$$U_4 = \mathbb{E}_{\mathbf{v}}[f_t(\mathbf{x}_t^* + \delta\mathbf{v}_t)] - f_t(\mathbf{x}_t^*) \leq \delta G. \quad (89)$$

Plugging (86)-(89) into (84), we arrive at

$$\begin{aligned}&\frac{1}{2}\sum_{t=1}^{T}\Big(\mathbb{E}[f_t(\mathbf{x}_{1,t})] + \mathbb{E}[f_t(\mathbf{x}_{2,t})]\Big) - \sum_{t=1}^{T} f_t(\mathbf{x}_t^*) \leq \frac{R}{\alpha}V(\mathbf{x}_{1:T}^*) \\ &+ \frac{R^2}{2\alpha} + 2\mu G^2R^2T + \alpha d^2G^2T + \gamma GRT(1+\|\bar{\boldsymbol{\lambda}}\|) + 2\delta GT. \quad (90)\end{aligned}$$

Upon choosing $\alpha = \mu = \mathcal{O}(T^{-\frac{1}{2}})$, and $\delta = \mathcal{O}(T^{-1})$ along with $\gamma = \delta/r$, it follows that (ignoring constant terms)

$$\mathrm{Reg}_T^{\mathrm{d}} = \mathcal{O}\Big(RV(\mathbf{x}_{1:T}^*)T^{\frac{1}{2}} + \frac{1}{2}R^2T^{\frac{1}{2}} + 2G^2R^2T^{\frac{1}{2}} + d^2G^2T^{\frac{1}{2}}\Big)$$

where we used the upper bound of dual variables in Lemma 7. This completes the proof of (26).

Dynamic fit in Theorem 2: To derive the bound on dynamic fit, recall that the constraint violations in (41) depend on the magnitude of the dual variable as well as on the difference of two consecutive primal iterates. The distance between iterates $\mathbf{x}_t$ and $\hat{\mathbf{x}}_{t+1}$ can be bounded by

$$\begin{aligned}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\| &\overset{(a)}{\leq} \left\|\alpha\hat{\nabla}_{\mathbf{x}}^2\mathcal{L}_t(\hat{\mathbf{x}}_t, \boldsymbol{\lambda}_t)\right\| \\ &\overset{(b)}{\leq} \|\hat{\nabla}^2 f_t(\hat{\mathbf{x}}_t)\| + \|\nabla\mathbf{g}_t(\hat{\mathbf{x}}_t)\|\|\boldsymbol{\lambda}_t\| \overset{(c)}{\leq} \alpha dG + \alpha G\|\bar{\boldsymbol{\lambda}}\| \quad (91)\end{aligned}$$

where (a) uses the non-expansive property of the projection operator, (b) applies the Cauchy-Schwarz inequality, and (c) relies on the bounds in (as2).

On the other hand, using the Lipschitz continuity of $\mathbf{g}_t(\mathbf{x})$ and (41), we have

$$\begin{aligned}&\frac{1}{2}\sum_{t=1}^{T}\left(\mathbf{g}_t(\mathbf{x}_{1,t}) + \mathbf{g}_t(\mathbf{x}_{2,t})\right) \leq \sum_{t=1}^{T}\mathbf{g}_t(\hat{\mathbf{x}}_t) + \delta GT\mathbf{1} \quad (92) \\ &\leq \frac{\boldsymbol{\lambda}_{T+1}}{\mu} + \frac{G^2T\mathbf{1}}{2\beta} + \frac{\beta}{2}\sum_{t=1}^{T}\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2\mathbf{1} + \delta GT\mathbf{1} \\ &\overset{(c)}{\leq} \frac{\boldsymbol{\lambda}_{T+1}}{\mu} + \frac{G^2T\mathbf{1}}{2\beta} + \beta T\Big(\alpha^2d^2G^2 + \alpha^2G^2\|\bar{\boldsymbol{\lambda}}\|^2\Big)\mathbf{1} + \delta GT\mathbf{1}\end{aligned}$$

where (c) uses (91), and the fact that $(a+b)^2 \leq 2(a^2+b^2)$.

In this case, if we take $[\cdot]^+$ and then $\|\cdot\|$ on both sides of (92), and further choose $\alpha = \mu = \mathcal{O}(T^{-\frac{1}{2}})$, $\delta = T^{-1}$, and $\beta = \mathcal{O}(T^{\frac{1}{2}})$, we arrive at

$$\begin{aligned}\mathrm{Fit}_T^{\mathrm{d}} \leq & \frac{\|\boldsymbol{\lambda}_{T+1}\|}{\mu} + \frac{G^2N^{\frac{1}{2}}T}{2\beta} + \beta N^{\frac{1}{2}}T\Big(\alpha^2d^2G^2 + \alpha^2G^2\|\bar{\boldsymbol{\lambda}}\|^2\Big) \\ = & CT^{\frac{1}{2}} + N^{\frac{1}{2}}T^{\frac{1}{2}}G^2\left(\frac{1}{2} + d^2 + C^2\right) = \mathcal{O}\left(T^{\frac{1}{2}}\right) \quad (93)\end{aligned}$$

where we used the bound on dual variables in Lemma 7. This completes also the proof of (28), and also that of Theorem 2.