# Online Convex Optimization for Dynamic Network Resource Allocation

Tianyi Chen*, Qing Ling†, and Georgios B. Giannakis*
*Dept. of Elec. & Comput. Engr. and Digital Technology Center, University of Minnesota, USA
†Dept. of Automation, University of Science and Technology of China, China

***Abstract*—The present paper deals with online convex optimization involving adversarial loss functions and adversarial constraints, where the constraints are revealed after making decisions, and can be tolerable to instantaneous violations but must be satisfied in the long term. Performance of an online algorithm in this setting is assessed by: i) the difference of its losses relative to the *best dynamic* solution with one-slot-ahead information of the loss function and the constraint (that is here termed *dynamic regret*); and, ii) the accumulated amount of constraint violations (that is here termed *dynamic fit*). In this context, a modified online saddle-point (MOSP) scheme is developed, and proved to simultaneously yield sub-linear dynamic regret and fit, provided that the accumulated variations of per-slot minimizers and constraints are sub-linearly growing with time. MOSP is applied to the dynamic network resource allocation task, and shown to outperform the well-known stochastic dual gradient method.**

***Index Terms*—Online convex optimization, online learning, non-stationary optimization, network resource allocation.**

## I. Introduction

Online convex optimization (OCO) is an emerging methodology for sequential inference with well documented merits especially when the sequence of convex costs varies in an unknown and possibly adversarial manner [2]. Starting from the seminal papers [2] and [3], most of the early works evaluate OCO algorithms with a *static regret*, which measures the difference of costs (a.k.a. losses) between the online solution and the overall best static solution in hindsight. However, static regret is not a comprehensive performance metric [4]. Take online parameter estimation as an example. When the true parameter varies over time, a static benchmark (time-invariant estimator) itself often performs poorly so that achieving sub-linear static regret is no longer attractive. Recent works [4]–[6] extend the analysis of static regret to that of *dynamic regret*, where the performance of an OCO algorithm is benchmarked by the best dynamic solution with a-priori information on the one-slot-ahead cost function. Sub-linear dynamic regret is proved to be possible, if the dynamic environment changes slow enough for the accumulated variation of either costs or per-slot minimizers to be sub-linearly increasing with respect to time.

The aforementioned works [4]–[6] deal with dynamic costs focusing on problems with time-invariant constraints that must be strictly satisfied, but do not allow for instantaneous violations of the constraints. The *long-term* effect of such instantaneous violations was studied in [7], where an online algorithm with sub-linear static regret and sub-linear accumulated constraint violation was also developed. Decentralized optimization with consensus constraints, as a special case of having long-term but time-invariant constraints, has been studied in [8], [9]. Nevertheless, [7]–[9] do not deal with OCO under time-varying adversarial constraints.

The present paper considers OCO with time-varying constraints. Relative to existing works, the main contributions of the present paper are summarized as follows.

**c1)** We generalize the standard OCO framework with only adversarial costs in [2]–[6] to account for both adversarial costs and constraints. Different from the static regret analysis in [7], [8], performance here is established relative to the best dynamic benchmark, via dynamic regret.

**c2)** We develop a modified online saddle-point (MOSP) algorithm for this novel OCO problem, which yields simultaneously sub-linear dynamic regret and fit, provided that the accumulated variations of per-slot minimizers and constraints are sub-linearly growing with time. This provides valuable insights: *When the dynamic environment comprising both costs and constraints does not change on average, the online decisions of MOSP are as good as the best dynamic solution over a long time horizon.*

**c3)** Our novel MOSP algorithm is further applied to dynamic resource allocation tasks, and compared with the popular stochastic dual gradient approach [10], [11]. Relative to the latter, MOSP remains operational in a broader setting without probabilistic assumptions. Simulations demonstrate remarkable performance gain of MOSP.

*Notation.* $(\cdot)^\top$ stands for vector and matrix transposition, and $\|\mathbf{x}\|$ denotes the $\ell_2$-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. OCO with Long-term Time-varying Constraints

In this section, we introduce the generic OCO formulation with long-term time-varying constraints, along with pertinent metrics to evaluate an OCO algorithm in this setting.

### A. Problem formulation

Online optimization with time-varying and long-term constraints is well motivated for applications such as traffic control, navigation, and network resource allocation [10]–[12]. Taking resource allocation as an example, time-varying long-term constraints are usually imposed to tolerate instantaneous violations when available resources cannot satisfy user requests, and hence allow flexible adaptation of online decisions to temporal variations of resource availability. To broaden the applicability of the classical OCO setting [2], [3] to these scenarios, we consider that per slot $t$, a learner selects an action $\mathbf{x}_t$ from a known and fixed convex set $\mathcal{X} \subseteq \mathbb{R}^I$, and then nature reveals

Work in this paper was supported by NSF 1509040, 1508993, 1509005, NSF China 61573331, NSF Anhui 1608085QF130, and CAS-XDA06040602. Proofs of results in this paper are available in the online version [1].

ISBN 978-0-9928626-7-1 © EURASIP 2017

not only a loss function $f_t(\cdot) : \mathbb{R}^I \to \mathbb{R}$ but also a time-varying (possibly adversarial) penalty function $\mathbf{g}_t(\cdot) : \mathbb{R}^I \to \mathbb{R}^I$. This function leads to a time-varying constraint $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}$, which is driven by the unknown dynamics in various applications, e.g., on-demand data request arrivals in resource allocation. Different from the known and fixed set $\mathcal{X}$, the time-varying constraint $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}$ can vary arbitrarily or even adversarially from slot to slot. It is revealed only after the learner makes her/his decision, and hence it is hard to be satisfied in every time slot. Therefore, the goal in this context is to find a sequence of online solutions $\{\mathbf{x}_t \in \mathcal{X}\}$ that minimize the aggregate loss, and ensure that the constraints $\{\mathbf{g}_t(\mathbf{x}_t) \leq \mathbf{0}\}$ are satisfied in the long term on average. Specifically, we aim to solve the following online optimization problem

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

where $T$ is the time horizon, $\mathbf{x}_t \in \mathbb{R}^I$ is the decision variable, $f_t$ is the cost function, $\mathbf{g}_t := [g_t^1, \ldots, g_t^I]^\top$ denotes the constraint function with $i$th entry $g_t^i : \mathbb{R}^I \to \mathbb{R}$, and $\mathcal{X} \in \mathbb{R}^I$ is a convex set. The formulation (1) extends the standard OCO framework [2]–[6] to accommodate adversarial time-varying constraints that must be satisfied in the long term. Complemented by algorithm development and performance analysis to be carried in the following sections, the main contribution of the present paper is incorporation of long-term and time-varying constraints to markedly broaden the scope of OCO.

### B. Performance and feasibility metrics

Regarding performance of online decisions $\{\mathbf{x}_t\}_{t=1}^T$, static regret is adopted as a metric by standard OCO schemes, under time-invariant and strictly satisfied constraints. The static regret measures the difference between the online loss of an OCO algorithm and that of the best fixed solution in hindsight [2], [3]. Extending the definition of static regret over $T$ slots to accommodate time-varying constraints, it can be written as $\mathrm{Reg}_T^{\mathrm{s}} := \sum_{t=1}^T f_t(\mathbf{x}_t) - \sum_{t=1}^T f_t(\mathbf{x}^*)$, where the best static solution $\mathbf{x}^*$ is obtained as $\mathbf{x}^* \in \arg\min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x})$ s. t. $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}, \forall t$. A desirable OCO algorithm in this case is the one yielding a sub-linear regret [7], meaning $\mathrm{Reg}_T^{\mathrm{s}} = \mathbf{o}(T)$. Consequently, $\lim_{T\to\infty} \mathrm{Reg}_T^{\mathrm{s}}/T = 0$ implies that the algorithm is "on average" no-regret, or in other words, asymptotically not worse than the best fixed solution $\mathbf{x}^*$. Though widely used in various OCO applications, the aforementioned *static regret* metric relies on a rather coarse benchmark, which may be less useful especially in dynamic settings. Furthermore, since the time-varying constraint $\mathbf{g}_t(\mathbf{x}_t) \leq \mathbf{0}$ is not observed before making a decision $\mathbf{x}_t$, its feasibility can not be checked instantaneously.

In response to the quest for improved benchmarks in this dynamic setup, two metrics are considered here: *dynamic regret* and *dynamic fit*. The notion of dynamic regret (also termed tracking regret or adaptive regret) has been recently introduced in [4]–[6] to offer a competitive performance measure of OCO algorithms under time-invariant constraints. We adopt it in the setting of (1) by incorporating time-varying constraints

$$\mathrm{Reg}_T^{\mathrm{d}} := \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}_t^*) \tag{2}$$

where 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

$$\mathbf{x}_t^* \in \arg\min_{\mathbf{x} \in \mathcal{X}} \ f_t(\mathbf{x}) \quad \text{s. t. } \mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}. \tag{3}$$

Clearly, the dynamic regret is always larger than the static regret, i.e., $\mathrm{Reg}_T^{\mathrm{s}} \leq \mathrm{Reg}_T^{\mathrm{d}}$, because $\sum_{t=1}^T f_t(\mathbf{x}^*)$ 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.

To ensure feasibility of online decisions, the notion of *dynamic fit* is introduced to measure the accumulated violation of constraints [7], [8]. It is defined as

$$\mathrm{Fit}_T^{\mathrm{d}} := \left\| \Big[ \sum_{t=1}^{T} \mathbf{g}_t(\mathbf{x}_t) \Big]^+ \right\|. \tag{4}$$

Observe that the dynamic fit is zero if the accumulated violation $\sum_{t=1}^T \mathbf{g}_t(\mathbf{x}_t)$ is entry-wise less than zero. However, enforcing $\sum_{t=1}^T \mathbf{g}_t(\mathbf{x}_t) \leq \mathbf{0}$ is different from restricting $\mathbf{x}_t$ to meet $\mathbf{g}_t(\mathbf{x}_t) \leq \mathbf{0}$ in each and every slot. While the latter readily implies the former, the long-term (aggregate) constraint allows adaptation of online decisions to the environment dynamics; as a result, it is tolerable to have $\mathbf{g}_t(\mathbf{x}_t) \geq \mathbf{0}$ and $\mathbf{g}_{t+1}(\mathbf{x}_{t+1}) \leq \mathbf{0}$.

An ideal algorithm in this broader OCO framework 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 [4].

## III. Modified Online saddle-point Method

In this section, a modified online saddle-point method is developed, and its performance and feasibility are analyzed. We show the novel approach generates a sequence $\{\mathbf{x}_t\}_{t=1}^T$ ensuring sub-linear dynamic regret and fit, under mild conditions of the cost and constraint variations.

### A. Algorithm development

Consider now the per-slot problem (3), which contains the current objective $f_t(\mathbf{x})$, the current constraint $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}$, and a time-invariant constraint set $\mathcal{X}$. With $\boldsymbol{\lambda} \in \mathbb{R}_+^N$ denoting the Lagrange multiplier associated with the time-varying constraint, the online (partial) Lagrangian of (3) can be expressed as

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

where $\mathbf{x} \in \mathcal{X}$ remains implicit. For the online Lagrangian (5), we introduce a modified online saddle-point (MOSP) approach, which takes a modified descent step in the primal domain followed by a dual ascent step at each time slot $t$. Specifically, given the previous primal iterate $\mathbf{x}_{t-1}$ and the current dual iterate $\boldsymbol{\lambda}_t$ at each slot $t$, the current decision $\mathbf{x}_t$ is the minimizer of the following optimization problem

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

where $\alpha$ is a positive stepsize, and $\nabla f_{t-1}(\mathbf{x}_{t-1})$ is the gradient of primal objective $f_{t-1}(\mathbf{x})$ at $\mathbf{x} = \mathbf{x}_{t-1}$. After the current decision $\mathbf{x}_t$ is made, $f_t(\mathbf{x})$ and $\mathbf{g}_t(\mathbf{x})$ are observed, and the dual update takes the form

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \mu \nabla_{\boldsymbol{\lambda}} \mathcal{L}_t(\mathbf{x}_t, \boldsymbol{\lambda}_t)\right]^+ = \left[\boldsymbol{\lambda}_t + \mu \mathbf{g}_t(\mathbf{x}_t)\right]^+ \quad (7)$$

where $\mu$ is also 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 online Lagrangian (5) w.r.t. $\boldsymbol{\lambda}$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}_t$.

**Remark 1.** The primal update of the classical saddle-point approach in [7], [8] is tantamount to minimizing a first-order approximation of $\mathcal{L}_{t-1}(\mathbf{x}, \boldsymbol{\lambda}_t)$ at $\mathbf{x} = \mathbf{x}_{t-1}$ plus a proximal term $\|\mathbf{x} - \mathbf{x}_{t-1}\|^2/(2\alpha)$. We call the primal-dual recursion (6) and (7) as a modified online saddle-point approach, since the primal update (6) is not an exact gradient step when the constraint $\mathbf{g}_t(\mathbf{x})$ is nonlinear w.r.t. $\mathbf{x}$. However, when $\mathbf{g}_t(\mathbf{x})$ is linear, (6) and (7) reduce to the approach in [7], [8]. The minimization in (6) penalizes the exact constraint violation $\mathbf{g}_t(\mathbf{x})$ instead of its first-order approximation, which improves control of constraint violations and facilitates performance analysis of MOSP.

### *B. Performance analysis*

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

**as1)** For every $t$, the cost function $f_t(\mathbf{x})$ and the time-varying constraint $\mathbf{g}_t(\mathbf{x})$ in (1) are convex.

**as2)** For every $t$, $f_t(\mathbf{x})$ has bounded gradient on $\mathcal{X}$; i.e., $\|\nabla f_t(\mathbf{x})\| \leq G$, $\forall \mathbf{x} \in \mathcal{X}$; and $\mathbf{g}_t(\mathbf{x})$ is bounded on $\mathcal{X}$; i.e., $\|\mathbf{g}_t(\mathbf{x})\| \leq M$, $\forall \mathbf{x} \in \mathcal{X}$.

**as3)** The radius of the convex feasible set $\mathcal{X}$ is bounded; i.e., $\|\mathbf{x} - \mathbf{y}\| \leq R$, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$.

**as4)** There exists a constant $\epsilon > 0$, and an interior point $\tilde{\mathbf{x}} \in \mathcal{X}$ such that $\mathbf{g}_t(\tilde{\mathbf{x}}) \leq -\epsilon \mathbf{1}$, $\forall t$.

Assumptions as1)-as2) are typical in OCO [3], [5], [8]; as3) restricts the action set to be bounded; as4) is Slater's condition, which guarantees the existence of a bounded Lagrange multiplier [13]. Under these assumptions, we are on track to first provide an upper bound for the dynamic fit.

**Theorem 1.** *Define the maximum variation of consecutive constraints as* $\bar{V}(\mathbf{g}) := \max_t V(\mathbf{g}_t)$*, with* $V(\mathbf{g}_t) := \max_{\mathbf{x} \in \mathcal{X}} \left\| [\mathbf{g}_{t+1}(\mathbf{x}) - \mathbf{g}_t(\mathbf{x})]^+ \right\|$*, and assume the slack constant* $\epsilon$ *in as4) to be larger than the maximum variation*[1]*; i.e.,* $\epsilon > \bar{V}(\mathbf{g})$*. Then under the dual variable initialization* $\boldsymbol{\lambda}_1 = \mathbf{0}$*, the dual iterate for the MOSP recursion* (6)-(7) *is bounded by*

$$\|\boldsymbol{\lambda}_t\| \leq \|\bar{\boldsymbol{\lambda}}\| := \mu M + \frac{2GR + R^2/(2\alpha) + (\mu M^2)/2}{\epsilon - \bar{V}(\mathbf{g})}, \ \forall t \quad (8)$$

*and the dynamic fit in* (4) *is upper-bounded by*

$$\mathrm{Fit}_T^{\mathrm{d}} \leq \frac{\|\bar{\boldsymbol{\lambda}}\|}{\mu} = M + \frac{2GR/\mu + R^2/(2\alpha\mu) + M^2/2}{\epsilon - \bar{V}(\mathbf{g})} \quad (9)$$

*where* $G$*,* $M$*,* $R$*, and* $\epsilon$ *are as in as2)-as4).*

Theorem 1 asserts that under a mild condition on the time-varying constraints, $\|\boldsymbol{\lambda}_t\|$ is uniformly upper-bounded, and more importantly, its scaled version $\|\boldsymbol{\lambda}_{T+1}\|/\mu$ upper bounds the dynamic fit. Observe that with a fixed primal stepsize $\alpha$, $\mathrm{Fit}_T^{\mathrm{d}}$ is in the order of $\mathcal{O}(1/\mu)$, thus a larger dual stepsize essentially enables a better satisfaction of long-term constraints. In addition, a smaller $\bar{V}(\mathbf{g})$ leads to a smaller dynamic fit, which also makes sense intuitively.

In the next theorem, we further bound the dynamic regret.

**Theorem 2.** *Under as1)-as4) and the dual variable initialization* $\boldsymbol{\lambda}_1 = \mathbf{0}$*, the MOSP recursion* (6)-(7) *yields*

$$\mathrm{Reg}_T^{\mathrm{d}} \leq \frac{RV(\{\mathbf{x}_t^*\}_{t=1}^T)}{\alpha} + \|\bar{\boldsymbol{\lambda}}\| V(\{\mathbf{g}_t\}_{t=1}^T) + \frac{R^2}{2\alpha} + \frac{\alpha G^2 T}{2} + \frac{\mu M^2 T}{2} \quad (10)$$

*where* $V(\{\mathbf{x}_t^*\}_{t=1}^T)$ *is the accumulated variation of the per-slot minimizers* $\mathbf{x}_t^*$ *defined as* $V(\{\mathbf{x}_t^*\}_{t=1}^T) := \sum_{t=1}^T \|\mathbf{x}_t^* - \mathbf{x}_{t-1}^*\|$ *and* $V(\{\mathbf{g}_t\}_{t=1}^T)$ *is the accumulated variation of consecutive constraints* $V(\{\mathbf{g}_t\}_{t=1}^T) := \sum_{t=1}^T V(\mathbf{g}_t)$.

Theorem 2 asserts that MOSP's dynamic regret is upper-bounded by a constant depending on the accumulated variations of per-slot minimizers and time-varying constraints as well as the primal and dual stepsizes. While the dynamic regret in the current form (10) is hard to grasp, the next corollary shall demonstrate that $\mathrm{Reg}_T^{\mathrm{d}}$ can be very small.

Based on Theorems 1-2, we are ready to establish that under the mild conditions, the dynamic regret and fit are sub-linearly increasing with $T$.

**Corollary 1.** *Consider as1)-as4) are satisfied, and the dual variable is initialized as* $\boldsymbol{\lambda}_1 = \mathbf{0}$*. If there exists a constant* $\beta \in [0, 1)$ *such that the temporal variations satisfy* $V(\{\mathbf{x}_t^*\}_{t=1}^T) = \mathcal{O}(T^\beta)$ *and* $V(\{\mathbf{g}_t\}_{t=1}^T) = \mathcal{O}(T^\beta)$*, then choosing the primal and dual stepsizes as* $\alpha = \mu = \mathcal{O}(T^{\frac{\beta-1}{2}})$ *leads to the dynamic fit* $\mathrm{Fit}_T^{\mathrm{d}} = \mathcal{O}(T^{1-\beta}) = \mathbf{o}(T)$ *and the corresponding dynamic regret* $\mathrm{Reg}_T^{\mathrm{d}} = \mathcal{O}(T^{\frac{\beta+1}{2}}) = \mathbf{o}(T)$.

Corollary 1 provides valuable insights for choosing stepsizes in non-stationary settings. Intuitively, when the variation of the environment is fast (a larger $\beta$), slowly decaying stepsizes (thus larger stepsizes) can better track the potential changes. We emphasize that sub-linear dynamic regret and fit in this novel OCO setting can be achieved when the dynamic environment consisting of the per-slot minimizer and the time-varying constraint *does not vary on average*, that is, $V(\{\mathbf{x}_t^*\}_{t=1}^T)$ and $V(\{\mathbf{g}_t\}_{t=1}^T)$ are sub-linearly increasing over $T$; i.e., $\beta < 1$.

## IV. Application to Network Resource Allocation

In this section, we solve the network resource allocation problem within the novel OCO framework, and present numerical experiments to demonstrate the merits of our MOSP solver.

### *A. Online network resource allocation*

Consider the resource allocation task over a cloud network [11], [14], which is represented by a directed graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ with node set $\mathcal{I}$ and edge set $\mathcal{E}$, where $|\mathcal{I}| = I$ and $|\mathcal{E}| = E$. Nodes include mapping nodes in the set $\mathcal{J} = \{1, \ldots, J\}$, and data centers in the set $\mathcal{K} = \{1, \ldots, K\}$; i.e., $\mathcal{I} = \mathcal{J} \bigcup \mathcal{K}$.

Per time $t$, each mapping node $j$ receives an exogenous data request $b_t^j$, and forwards the amount $x_t^{jk}$ to each data center $k$ in accordance with bandwidth availability. Each data

[1]This equivalently requires $\epsilon := \min_{i,t} \max_{\mathbf{x} \in \mathcal{X}} [-g_t^i(\mathbf{x})]^+ > \max_{\mathbf{x} \in \mathcal{X}} \left\| [\mathbf{g}_{t+1}(\mathbf{x}) - \mathbf{g}_t(\mathbf{x})]^+ \right\|$, which is valid when the region defined by $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}$ is large enough, or, the trajectory of $\mathbf{g}_t(\mathbf{x})$ is smooth enough across time.

ISBN 978-0-9928626-7-1 © EURASIP 2017

center $k$ schedules workload $y_t^k$ according to its resource availability. Regarding $y_t^k$ as the workloads of a virtual outgoing edge $(k,*)$ from data center $k$, edge set $\mathcal{E} := \{(j,k), \forall j \in \mathcal{J}, k \in \mathcal{K}\} \bigcup \{(k,*), \forall k \in \mathcal{K}\}$ contains all the links connecting mapping nodes with data centers, and all the "virtual" edges coming out of the data centers. The $I \times E$ node-incidence matrix is formed with the $(i,e)$-th entry $\mathbf{A}_{(i,e)} = 1$ if link $e$ enters node $i$; $\mathbf{A}_{(i,e)} = -1$, if link $e$ leaves node $i$; and $\mathbf{A}_{(i,e)} = 0$, otherwise. For compactness, collect the workloads across each edge $e = (i,j) \in \mathcal{E}$ in a resource allocation vector $\mathbf{x}_t := [x_t^{11}, \ldots, x_t^{JK}, y_t^1, \ldots, y_t^K]^\top \in \mathbb{R}_+^E$, and the exogenous load arrival rates of all nodes in a vector $\mathbf{b}_t := [b_t^1, \ldots, b_t^J, 0 \ldots, 0]^\top \in \mathbb{R}_+^I$. Then, the aggregate (endogenous plus exogenous) workloads of all nodes are given by $\mathbf{A}\mathbf{x}_t + \mathbf{b}_t$. Assume that each data center and mapping node has a local data queue to buffer unserved workloads [10]. With $\mathbf{q}_t := [q_t^1, \ldots, q_t^{J+K}]^\top$ collecting their queue lengths, the queue update is $\mathbf{q}_{t+1} = [\mathbf{q}_t + \mathbf{A}\mathbf{x}_t + \mathbf{b}_t]^+$. The bandwidth limit of link $(j,k)$ is $\bar{x}^{jk}$, and the capability of data center $k$ is $\bar{y}^k$, which can be compactly expressed by $\mathbf{x} \in \mathcal{X}$ with $\mathcal{X} := \{\mathbf{0} \leq \mathbf{x} \leq \bar{\mathbf{x}}\}$ and $\bar{\mathbf{x}} := [\bar{x}^{11}, \ldots, \bar{x}^{JK}, \bar{y}^1, \ldots, \bar{y}^K]^\top$; see the diagram in Fig. 1.

For each data center, the power cost $f_t^k(y_t^k) := f^k(y_t^k; \theta_t^k)$ depends on a time-varying parameter $\theta_t^k$, which captures the energy price and the renewable generation at data center $k$ during slot $t$. The bandwidth cost $f_t^{jk}(x_t^{jk}) := f^{jk}(x_t^{jk}; \theta_t^{jk})$ characterizes the transmission delay and is parameterized by a scalar $\theta_t^{jk}$. Per slot $t$, the cost $f_t(\mathbf{x}_t)$ aggregates the power costs at all data centers plus the bandwidth costs at all links, namely

$$f_t(\mathbf{x}_t) := \sum_{k\in\mathcal{K}} \underbrace{f_t^k(y_t^k)}_{\text{power cost}} + \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} \underbrace{f_t^{jk}(x_t^{jk})}_{\text{bandwidth cost}} \quad (11)$$

where the objective can be also written as $f_t(\mathbf{x}_t) := f(\mathbf{x}_t; \boldsymbol{\theta}_t)$ with $\boldsymbol{\theta}_t := [\theta_t^1, \ldots, \theta_t^K, \theta_t^{11}, \ldots, \theta_t^{JK}]^\top$ concatenating all time-varying parameters. Aiming to minimize the accumulated cost while serving all workloads, the optimal workload routing and allocation strategy in this cloud network is the solution of the following optimization problem

$$\min_{\{\mathbf{x}_t \in \mathcal{X}, \forall t\}} \sum_{t=1}^{T} f_t(\mathbf{x}_t) \;\; \text{s. t.} \;\; \mathbf{q}_{t+1} = [\mathbf{q}_t + \mathbf{A}\mathbf{x}_t + \mathbf{b}_t]^+, \forall t \quad (12)$$

where $\mathbf{q}_1 \geq \mathbf{0}$ is the given initial queue length, and $\mathbf{q}_{T+1} = \mathbf{0}$ guarantees that all workloads arrived have been served at the end of the scheduling horizon. Note that (12) is time-coupled, and generally challenging to solve without information of future workload arrivals and time-varying cost functions. Relaxing the queue recursion in (12) by $\mathbf{q}_{T+1} \geq \mathbf{q}_1 + \sum_{t=1}^{T}(\mathbf{A}\mathbf{x}_t + \mathbf{b}_t)$, it readily leads to $\sum_{t=1}^{T}(\mathbf{A}\mathbf{x}_t + \mathbf{b}_t) \leq \mathbf{q}_{T+1} - \mathbf{q}_1 \leq \mathbf{0}$, since $\mathbf{q}_1 \geq \mathbf{0}$ and $\mathbf{q}_{T+1} = \mathbf{0}$. Therefore, instead of solving (12), we aim to tackle a relaxed problem that is in the form of OCO with long-term constraints, given by

$$\min_{\{\mathbf{x}_t \in \mathcal{X}, \forall t\}} \sum_{t=1}^{T} f_t(\mathbf{x}_t) \quad \text{s. t.} \quad \sum_{t=1}^{T} (\mathbf{A}\mathbf{x}_t + \mathbf{b}_t) \leq \mathbf{0} \quad (13)$$

where the workload flow conservation constraint $\mathbf{A}\mathbf{x}_t + \mathbf{b}_t \leq \mathbf{0}$ must be satisfied in the long term rather than slot-by-slot. Clearly, (13) is in the form of (1). Therefore, the MOSP algorithm of Section III can be leveraged to solve (13) in an *online* fashion, with provable performance and feasibility guarantees. Specifically, with $\mathbf{g}_t(\mathbf{x}_t) = \mathbf{A}\mathbf{x}_t + \mathbf{b}_t$, the primal update (6) boils down to a simple gradient update $\mathbf{x}_t = \mathcal{P}_{\mathcal{X}}\left(\mathbf{x}_{t-1} - \alpha\nabla f_{t-1}(\mathbf{x}_{t-1}) - \alpha\mathbf{A}^\top\boldsymbol{\lambda}_t\right)$, where $\mathcal{P}_{\mathcal{X}}(\cdot)$ defines projection onto the convex set $\mathcal{X}$. The dual update (7) is $\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t + \mu(\mathbf{A}\mathbf{x}_t + \mathbf{b}_t)\right]^+$, which can be nicely regarded as a scaled version of the queue dynamics in (12), with $\mathbf{q}_t = \boldsymbol{\lambda}_t/\mu$.

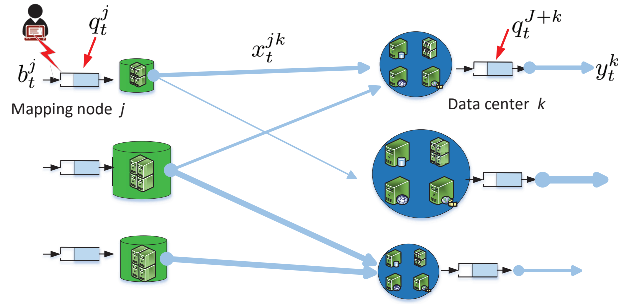



Fig. 1: A diagram of online network resource allocation. Per time $t$, mapping node $j$ has an exogenous workload $b_t^j$ plus that stored in the queue $q_t^j$, and schedules workload $x_t^{jk}$ to data center $k$. Data center $k$ serves an amount of workload $y_t^k$ out of the assigned $\sum_{j=1}^{J} x_t^{jk}$ as well as that in its queue $q_t^{J+k}$.

In addition to simple closed-form updates, MOSP can also afford a fully decentralized implementation, where each mapping node or data center decides the amounts on all its *outgoing links*, and only exchanges information with its *one-hop neighbors*. Per time slot $t$, the primal update at mapping node $j$ includes variables on all its outgoing links, given by

$$x_t^{jk} = \left[x_{t-1}^{jk} - \alpha\nabla f_{t-1}^{jk}(x_{t-1}^{jk}) - \alpha\left(\lambda_t^k - \lambda_t^j\right)\right]_0^{\bar{x}^{jk}}, \forall k \in \mathcal{K} \quad (14a)$$

and the dual update reduces to

$$\lambda_{t+1}^j = \left[\lambda_t^j + \mu\Big(b_t^j - \textstyle\sum_{k\in\mathcal{K}} x_t^{jk}\Big)\right]^+. \quad (14b)$$

Likewise, for data center $k$, the primal update becomes

$$y_t^k = \left[y_{t-1}^k - \alpha\nabla f_{t-1}^k(y_{t-1}^k) - \alpha\textstyle\sum_{j\in\mathcal{J}}(\lambda_t^k - \lambda_t^j)\right]_0^{\bar{y}^k} \quad (14c)$$

where $[\,\cdot\,]_0^{\bar{y}^k} := \min\{\bar{y}^k, \max\{\cdot, 0\}\}$, and the dual recursion is

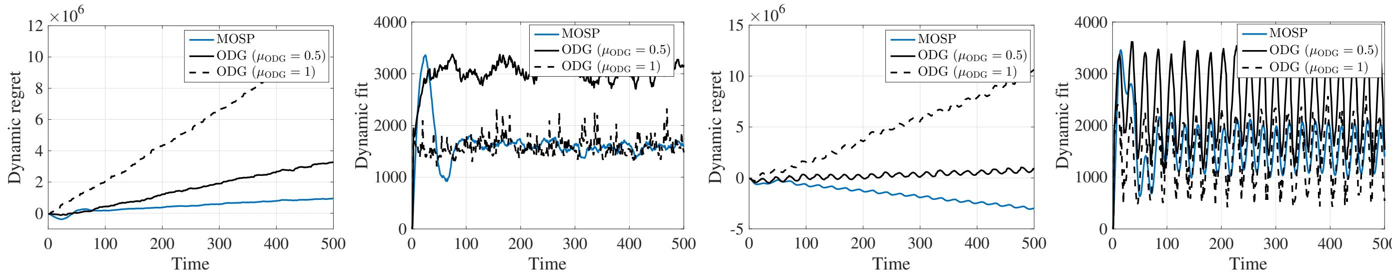
$$\lambda_{t+1}^k = \left[\lambda_t^k + \mu\Big(\textstyle\sum_{j\in\mathcal{J}} x_t^{jk} - y_t^k\Big)\right]^+. \quad (14d)$$

### *B. Numerical experiments*

Consider the resource allocation task in (13) with $J = 10$ mapping nodes and $K = 10$ data centers. The cost in (11) is $f_t(\mathbf{x}_t) := \sum_{k\in\mathcal{K}} p_t^k (y_t^k)^2 + \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} c^{jk}(x_t^{jk})^2$, where $p_t^k$ is the energy price of data center $k$ at time $t$, and $c^{jk}$ is the per-unit bandwidth cost for transmitting from mapping node $j$ to data center $k$. With the bandwidth limit $\bar{x}^{jk}$ uniformly randomly generated within $[10, 100]$, we set the bandwidth cost of each link $(j,k)$ as $c^{jk} = 40/\bar{x}^{jk}$. The resource capacities $\{\bar{y}^k, \forall k\}$ are uniformly randomly generated from $[100, 200]$. Two cases are considered for the parameters $\{p_t^k, \forall t, k\}$ and $\{b_t^j, \forall t, j\}$:

**Case 1)** Parameters are *independently* drawn from invariant distributions. Specifically, $p_t^k$ is uniformly distributed over $[1, 3]$, and the delay-tolerant workload $b_t^j$ arrives at each mapping node $j$ according to a uniform distribution over $[50, 150]$.

ISBN 978-0-9928626-7-1 © EURASIP 2017




(a) Dynamic regret for Case 1. (b) Dynamic fit for Case 1. (c) Dynamic regret for Case 2. (d) Dynamic fit for Case 2.

Fig. 2: Online performance of MOSP under Cases 1 and 2.

**Case 2)** Parameters are generated according to *non-stationary* stochastic processes. Specifically, $p_t^k = \sin(\pi t/12) + n_t^k$ with i.i.d. noise $n_t^k$ uniformly distributed over $[1, 3]$, while $b_t^j = 50\sin(\pi t/12) + v_t^j$ with $v_t^j$ uniformly distributed over $[99, 101]$.

Finally, with $T = 500$, the stepsize in (14a) and (14c) is set to $\alpha = 0.05/T^{1/3}$, and for (14b) and (14d) to $\mu = 50/T^{1/3}$. MOSP is benchmarked by three strategies: the popular stochastic dual gradient (SDG) approach in [10], [11], the sequence of per-slot best minimizers in (3), and the offline optimal solution that solves (1) at once with all future costs and constraints available. Note that at the beginning of each slot $t$, the exact prices $\{p_t^k, \forall k\}$ and demands $\{b_t^j, \forall j\}$ for the coming slot are generally not available in practice [15]. Since the SDG updates in [10] require non-causal knowledge of $\{p_t^k, \forall k\}$ and $\{b_t^j, \forall j\}$ to decide $\mathbf{x}_t$, we modify them in this online setting by using the prices and demands at slot $t-1$ to obtain $\mathbf{x}_t$. In this case that we term online dual gradient (ODG), the performance guarantee of SDG may not hold. Nevertheless, different constant stepsizes for ODG's dual update still lead to quite different behaviors, thus ODG is studied under stepsizes $\mu_{\rm ODG} = 0.5$ and 1.

Figs. 2a-2b show the test results for Case 1 under i.i.d. costs and constraints. The dynamic regret (cf. (2)) of MOSP grows much slower than that of ODG in Fig. 2a. Regarding the dynamic fit (cf. (4)), Fig. 2b demonstrates that ODG with $\mu_{\rm ODG} = 1$ has a smaller fit than that of $\mu_{\rm ODG} = 0.5$, and similar to the dynamic fit of MOSP. According to the well-known trade-off between cost (optimality) and delay (constraint violations) in [10], increasing $\mu_{\rm ODG}$ will improve the dynamic fit of ODG but degrade its dynamic regret. Therefore, MOSP is favorable in Case 1 since it has much smaller regret when its dynamic fit is similar to that of ODG with $\mu_{\rm ODG} = 1$. Different from Case 1, the dynamic regret of MOSP in Case 2 is not only much smaller than ODG, but also smaller than the per-slot optimum obtained via (3); see the negative regret of Fig. 2c. Note that the dynamic regret can be negative since $\mathbf{g}_t(\mathbf{x}) \leq \mathbf{0}$ is not strictly satisfied by MOSP's solutions per-slot [1, Sec. III.C]. Regarding Fig. 2d, accumulated constraint violations of both ODG and MOSP do not increase with time. The dynamic fit of MOSP is much smaller than that of ODG with $\mu_{\rm ODG} = 0.5$, and comparable to that of ODG with $\mu_{\rm ODG} = 1$. Therefore, in this non-stationary case, MOSP also outperforms ODG.

## V. Concluding Remarks

OCO with both adversarial costs and constraints has been studied in this paper. Different from existing works, the focus is on a setting where some of the constraints are revealed after taking actions, tolerable to instantaneous violations but having to be satisfied on average. Performance of the novel OCO algorithm is measured by: i) the difference of its objective relative to the best dynamic solution with one-slot-ahead information of the cost and the constraint (dynamic regret); and, ii) its accumulated amount of constraint violations (dynamic fit). For network resource allocation, it has been shown that MOSP simultaneously yields sub-linear dynamic regret and fit, if the accumulated variations of the dynamic solutions and constraints are sub-linearly growing with time. This novel OCO setting broaden the applicability of OCO to a wider application regime, which includes, e.g., online demand response in smart grids.

## References

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

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

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Trans. Signal Processing*, vol. 63, no. 19, pp. 5149–5164, Oct. 2015.

[9] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *arXiv preprint:1609.02845*, Sep. 2016.

[10] 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.

[11] T. Chen, A. G. Marques, and G. B. Giannakis, "DGLB: Distributed stochastic geographical load balancing over cloud networks," *IEEE Trans. Parallel and Distrib. Syst.*, to appear, 2017.

[12] J. Gregoire, X. Qian, E. Frazzoli, A. de La Fortelle, and T. Wongpiromsarn, "Capacity-aware backpressure traffic signal control," *IEEE Trans. Control of Network Systems*, vol. 2, no. 2, pp. 164–173, June 2015.

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

[14] 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*, May 2017, to appear. [Online]. Available: https://arxiv.org/abs/1610.02143

[15] S. J. Kim and G. Giannakis, "An online convex optimization approach to real-time energy pricing for demand response," *IEEE Trans. Smart Grid*, to appear, 2017.