Online Convex Optimization with Switching Costs: Algorithms and Performance

Qingsong Liu*, Zhuoran Li* and Zhixuan Fang*†
*IIIS, Tsinghua University, Beijing, China
†Shanghai Qi Zhi Institute, Shanghai, China

***Abstract*—In this paper, we study the problem of online convex optimization with switching costs (SOCO) that appears in diverse scenarios including power management, video streaming, resource allocation, etc. SOCO refers to online decision problems when the agent incurs both hitting cost and an additional switching cost of changing decisions. We adopt the universal dynamic regret as the performance metric, and consider the case when loss functions are unknown when making decisions. Previous results rely on the assumptions on loss function, e.g., linearity and smoothness, or prior knowledge of regularity measures, e.g., path length of the comparator sequence. In this paper, we propose an algorithm IOMD-SOCO that applies to general convex loss function, and show that the algorithm achieves an order-optimal universal dynamic regret bound. We also propose its parameter-free versions, i.e., without requiring the prior knowledge of path length of the comparator sequence, and achieve the same-order regret bound. We are the first to provide dynamic regret bounds for SOCO with general convex loss functions via parameter-free algorithm. Our numerical experiments show that IOMD-SOCO indeed achieves a substantial performance improvement.**

I. Introduction

Online Convex Optimization with switching costs (SOCO) is a variant of the classic Online Convex Optimization (OCO) [26], [29], where the agent (online learner) incurs a switching cost for changing its decisions. Due to the model's generality, SOCO captures problems in many scenarios such as cloud resource provisioning [10], online portfolio optimization with transaction costs [21], electric vehicle charging [14], etc.

Specifically, SOCO could be viewed as a game between an agent and an adaptive adversary or, the environment, performed in a sequence of consecutive rounds. At round t, after submitting a decision x_t chosen from the decision set χ, the agent receives a loss $f_t(x_t)$ and incurs an additional cost $c(x_t, x_{t-1})$ that penalizes the change of the consecutive decisions. The objective of the agent is to minimize its total cost over T rounds:

$$\text{Cost}(T) = \sum_{t=1}^{T} f_t(x_t) + c(x_t, x_{t-1}).$$

Since we assume f_t is revealed after making the decision x_t, the SOCO problem we considered belongs to the online learning formulation, in which the literature usually adopts regret as the performance metric. Thus in our paper, we use the regret to evaluate the performance of agent, which is defined as the difference between the total cost incurred by the agent and that of a given comparator sequence $u_0, ..., u_T \in \chi$:

Corresponding author: Zhixuan Fang at IIIS, Tsinghua University (Email: zfang@mail.tsinghua.edu.cn).

$$\begin{aligned}&\text{Regret}(u_{1:T})\\ &= \sum_{t=1}^{T} f_t(x_t) + c(x_t, x_{t-1}) - \sum_{t=1}^{T} (f_t(u_t) + c(u_t, u_{t-1})).\end{aligned} \tag{1}$$

Note that the regret form we consider is general, in which the sequence of comparators $\{u_t\}_{t=1}^{T}$ can be arbitrarily given, i.e., the universal dynamic regret. This form of regret has attracted an increasing attention in the recent literature of OCO [25], [29] due to its generality. Regrets in previous studies for SOCO are all special cases of (1). For example, most of the existing studies which considered the polynomial l_2 norm switching costs (e.g., [8], [10], [11], [17]) used the restricted dynamic regret. Meanwhile, some very recent work (e.g., [3]) considers the universal static regret, which is also our special case.

An ideal online algorithm is to choose x_t at each round t to obtain a sublinear growth of regret with respect to the time horizon T. And we explore the analytical performance in terms of universal dynamic regret. Note that the performance guarantee with respect to the dynamic regret of any online algorithm is strongly correlated with the temporal variations of the dynamic environment and comparator sequence. Intuitively, the regret bound of any online algorithm should depend on how drastically the loss function f_t and the comparator u_t vary across time. Below, we introduce two kinds of commonly-adopted regularities on the temporal variations of dynamic environment and comparator sequence from the literature of OCO (e.g., [2], [25], [28], [29]).

- Path-length (e.g., [25], [28], [29]):
$$P_T = \sum_{t=1}^{T} ||u_t - u_{t-1}||_p,\ p \geq 1.$$
- Function-variation (e.g., [2], [25], [28]):
$$V_f = \sum_{t=1}^{T} \max_{x \in \chi} |f_t(x) - f_{t-1}(x)|.$$

Here we let $u_0 = f_0 = 0$. These regularities are not comparable in general and are favored in different scenarios [26], [28]. It is known that in the worst case, it is impossible to achieve a sublinear dynamic regret bound for any online algorithm unless regularity measures are sublinear [26].

Previous work on SOCO mostly focuses on switching cost of l_1 norm [21] and l_2 norm [6], [10], [24], as well as squared l_2 norm [11], [12], [17]. Recently the squared l_2 norm switching cost has attracted a lot of attention due to

its importance in many practical scenarios such as online regression [11], economic dispatch in power systems [17], trajectory tracking in LTV systems [17], and dynamic resource management in data centers [10]. Our present paper considers switching costs of the squared l_p ($p \geq 1$) norm, i.e., $c(x_t, x_{t-1}) = \frac{1}{2}||x_t - x_{t-1}||_p^2$ (without loss of generality, we let the coefficient be $\frac{1}{2}$). Note that the form of switching costs we consider generalizes the squared l_2 norm switching costs. The squared norm switching cost has several applications to diverse problems across learning and control, e.g., LQR control [12] ($p = 2$), and smoothed online regression [12] in which the regularization term is square l_1 or l_2 norm distance.

In this paper, we follow the line of research stemmed from the advanced algorithm R-OBD [11] for SOCO with predictable loss functions. We present its variant, named IOMD-SOCO. By adopting the same spirit of IOMD in [5], and designing appropriate time-varying. learning parameters to address the SOCO problem with squared l_p norm switching costs. We also propose the parameter-free versions of IOMD-SOCO, and investigate their performance in terms of universal dynamic regret for general convex loss functions.

A. Contributions

We summarize our main results as follows:

- We propose the IOMD-SOCO algorithm and show that it enjoys an upper bound of $O(\min\{V_f, \sqrt{T + TP_T}\})$ on the universal dynamic regret for generally convex loss functions. We also provide lower bounds to show that this dynamic regret bound is tight (order-optimal).
- When the path-length P_T is unknown in advance, we adopt and propose the online ensemble method to tune the learning parameters in IOMD-SOCO and proposed a parameter-free algorithm Hedge-IOMD-SOCO. Our theoretical result shows that it can achieve a regret bound of $O(\min\{\sqrt{T + TP_T}, \max\{V_f, \sqrt{T}\}\})$, which only incurs a slightly worse performance degradation compared to IOMD-SOCO.
- When the path length P_T could be calculated on the fly, we design another parameter-free version of IOMD-SOCO, AdaIOMD-SOCO, and show that it could preserve the regret bound of IOMD-SOCO (i.e., $O(\min\{V_f, \sqrt{T + TP_T}\})$).

B. Related Work

In this subsection, we mainly review the literature of SOCO with squared norm switching costs, which is mostly related to ours. We categorize these work by whether the agent can observe the loss functions before making decisions.

Known loss functions. There are several works on SOCO with a single-step prediction of loss functions, that is, the loss function f_t is known to the agent before making the decision at round t. [12] first studied OCO with squared l_2 norm switching cost given strongly-convex loss functions. They proposed Online Balanced Descent (OBD) algorithm, which achieves a constant competitive ratio. Later [11] proposed two improved algorithms, G-OBD and R-OBD, and proved that R-OBD can achieve sublinear dynamic regret. [30] showed that by applying the standard Online Mirror Descent (OMD), one can achieve a sublinear dynamic regret for smooth loss functions when switching cost is l_2 norm or squared l_2 norm.

Moreover, recently, [17] studied the SOCO with squared l_2 norm switching costs under multi-step prediction setting, that is, the agent can observe the next W loss functions $\{f_t, ..., f_{t+W-1}\}$ before making decision x_t. They developed online algorithms based on horizon gradient descent (HGD) and established an $O(V_\theta)$ restricted dynamic regret bound, where V_θ is the path-length of sequence $\{\theta_t\}_{t=1}^T$ and $\theta_t = \arg\min_{x \in \chi} f_t(x)$. However, their results rely on the strong assumption that the loss functions are strongly-convex and smooth. [22] also considered squared l_2 norm switching costs with multi-step predictions of loss functions. When the prediction window is one, their setting could reduces to the same setting in our paper. However, they studied the competitive ratio for their algorithms' performance, and didn't provide results on regret. Besides, prediction is usually considered as imperfect in previous study (e.g., [7], [15]). When the prediction window is limited to one and the prediction error is zero, the setting of [15] reduces to the one considered in our paper. In this case, their algorithms reduce to OGD algorithm and achieve a restricted dynamic regret bound of $\sqrt{TV_f}$.

Unknown loss functions. When the agent does not know f_t and any future loss functions at the time of making decision x_t, i.e., online learning setting. [30] studied SOCO with squared l_2 norm switching cost without prediction and developed an algorithm based on the OMD method and confirm its sublinear dynamic regret guarantee for smooth loss functions. This work is mostly related to our work since we will also study the SOCO with squared norm switching costs without prediction and choose dynamic regret as the performance metric. Very recently, [3] also study the SOCO problems with squared l_2 norm switching costs without prediction, but they use the static benchmark and assume the loss functions are linear.

We summarize our and previous relevant results in Table I.

C. Preliminaries

We denote $D_\Phi : \chi \times \chi \to R^+$ the Bregman Divergence w.r.t. Φ, which is defined as $D_\Phi(x, y) = \Phi(x) - \Phi(y) - \langle \nabla\Phi(y), x - y\rangle$. For convenience, we let $||\cdot||$ be the l_p norm throughout the rest of paper, and assume that Φ is strongly convex w.r.t $||\cdot||$ in χ. Without loss of generality, we could assume the strong convexity constant of Φ to be $\frac{1}{2}$, then we have $D_\Phi(x, y) \geq \frac{1}{2}||x - y||^2$. The symbol $O(\cdot)$ hides the constants without affecting the final regret order.

We assume that the following assumption holds throughout the paper, which is a common assumption in the literature of OCO [25], [29].

Assumption 1: We make following assumptions w.r.t feasible set χ, loss functions $f_1, f_2..., f_T$:

- The feasible set χ is closed, convex and compact. $\forall x, y \in \chi$, it holds that

$$\max\{||x - y||, \sqrt{2D_\Phi(x, y)}\} \leq R.$$

Table I. Results on regret with polynomial norm switching cost. In this table, L represents the path-length or budget of the path-length, and the symbol $O(\cdot)$ hides the $\sqrt{T}$ term since $L \geq O(1)$. Parameter-free means the parameter setting in the corresponding algorithm does not require the prior knowledge of L and V_f. The metric refers to the regret form they adopt, and dynamic regrets in previous literature are special cases of the universal dynamic regret. The Knowledge of f_t reflects whether f_t is known at the beginning of round t.

Knowledge	Algorithm	Assumption on f_t	Switching costs	Metric	Parameter-free	Upper bounds
f_t is known	OBD [8]	locally-polyhedral, convex	$\|\|x_t - x_{t-1}\|\|_2$	dynamic	✗	$O(\sqrt{TL})$
	OBD [12]	strongly-convex, smooth	$\|\|x_t - x_{t-1}\|\|_2^2$		✗	$O(T)$
	R-OBD [11]	strongly-convex	$\|\|x_t - x_{t-1}\|\|^2$		✗	$O(\sqrt{TL})$
	OBD, RHC [10]	convex	$\|\|x_t - x_{t-1}\|\|_2$		✗	$O(\sqrt{TL})$
	OMD [30]	convex, smooth	$1 \leq \sigma \leq 2$ $\|\|x_t - x_{t-1}\|\|_2^\sigma$		✗	$O(\sqrt{TL})$
	RHGD, RHAG [17]	strong-convex, smooth	$\|\|x_t - x_{t-1}\|\|_2^2$		✗	$O(\sqrt{TL})$
	RHIG [15]	strong-convex, smooth	$\|\|x_t - x_{t-1}\|\|^2$		✗	$O(\sqrt{TV_f})$
f_t is unknown	OGD [10], [24]	convex	$\|\|x_t - x_{t-1}\|\|_2$	dynamic	✗	$O(\sqrt{TL})$
	OMD [30]	convex, smooth	$1 \leq \sigma \leq 2$ $\|\|x_t - x_{t-1}\|\|_2^\sigma$	dynamic	✗	$O(\sqrt{TL})$
	[3]	linear	$\|\|x_t - x_{t-1}\|\|_2^{1+\sigma}$ $\sigma \geq 0$	(universal) static	✓	$O((\log T)^{\frac{1+\min\{1,\sigma\}}{2}}$ $T^{\frac{1-\min\{1,\sigma\}}{2}})$
	OGDM [21]	convex	$\|\|x_t - x_{t-1}\|\|_1$	static	✓	$O(\sqrt{T})$
f_t is unknown (This paper)	IOMD-SOCO	convex	$\|\|x_t - x_{t-1}\|\|^2$	(universal) dynamic	✗	$O(\min\{V_f, \sqrt{TL}\})$
	IOMD-SOCO	convex	$\|\|x_t - x_{t-1}\|\|^2$		✓	$O(\min\{P_T V_f, \sqrt{T} P_T\})$
	Hedge-IOMD-SOCO	convex	$\|\|x_t - x_{t-1}\|\|^2$		✓	$O(\min\{\sqrt{TL}, \max\{V_f, \sqrt{T}\}\})$
	AdaIOMD-SOCO	convex	$\|\|x_t - x_{t-1}\|\|^2$		✓	$O(\min\{V_f, \sqrt{TL}\})$

Algorithm 1 IOMD-SOCO

1: **Initialize**: x_1, β, $\lambda_0 = 1$.
2: **for** round $t = 1...T$ **do**
3: $\lambda_t = \frac{1}{\beta^2}\sum_{i=1}^{t-1}\delta_i + 1$ (or $\lambda_t = \lambda_{t-1} + \frac{\delta_{t-1}}{\beta^2}$)
4: $x_{t+1} = \arg\min_{x\in\chi} f_t(x) + \lambda_t D_\Phi(x, x_t)$
5: $\delta_t = f_t(x_t) - f_t(x_{t+1}) - \lambda_t D_\Phi(x_{t+1}, x_t) + \frac{1}{2}||x_{t+1} - x_t||^2$
6: Choose action x_{t+1} and then observe f_{t+1}
7: **end for**

- Loss functions $f_1...f_T$ are convex, and there exists a constant $G > 0$ such that

$$||\nabla f_t(x)|| \leq G, \forall x \in \chi, t.$$

Here it also implies that f_t is Lipschitz continuous with parameter G.

In this paper, all the proofs of listed lemmas and theorems are given in our online report [1].

II. Main Results

In this section, we propose our algorithm, IOMD-SOCO for SOCO with squared l_p norm switching costs (see Algorithm 1), and provide its regret guarantee in the following theorem.

Theorem 1: Under Assumption 1, let $\beta^2 = R^2 + RP_T$, then IOMD-SOCO ensures that

$$\begin{aligned}\text{Regret}(u_{1:T}) &\leq R^2\lambda_T + R\sum_{t=2}^{T}\lambda_t||u_t - u_{t-1}|| + \sum_{t=1}^{T}\delta_t \\ &\leq O(\min\{V_f, \sqrt{T + TP_T}\}),\end{aligned} \tag{2}$$

where $\delta_t = f_t(x_t) - f_t(x_{t+1}) - \lambda_t D_\Phi(x_{t+1}, x_t) + \frac{1}{2}||x_{t+1} - x_t||^2$.

The regret bound in Theorem 1 outperforms any of prior results (e.g., [11], [12], [30]) in terms of dynamic regret, whether they assumed the loss functions are known in advance or not (See table I). For a group of practical SOCO problems such as economic dispatch in power systems [16], in which the loss function remains relatively stable, i.e., $V_f = O(1)$, our algorithm IOMD-SOCO can get a constant regret. As far as we know, there is no online algorithm in the literature that could achieve a constant regret for any convex loss function sequence when $V_f = O(1)$, even if the loss functions could be observed before making decisions. Besides, in many practical scenarios wherein f_t is linear and i.i.d across time (i.e., $V_f = O(\sqrt{T})$), our regret bound can also greatly outperform the existing works. In general, our regret bound is superior when the variation of consecutive loss functions is smooth across time.

Here we discuss how the adaption of different norms affects the learning performance of IOMD-SOCO. Note that the path-length P_T we adopted is the l_p norm and if we use the standard l_2 norm P_T, the performance bound would incur a multiplicative factor $n^{\frac{1}{p}-\frac{1}{2}}$ since $||\cdot||_p \leq n^{\frac{1}{p}-\frac{1}{2}}||\cdot||_2$, where n is the dimension.

Remark 1: Note that the standard analysis from previous work (e.g., OGD, OMD) cannot achieve our regret bound. For example, although OGD (OMD) ends up moving about $O(\frac{1}{t})$ in the t-th iteration, it can only yield $O(\sqrt{TP_T + T} + \log T)$ regret bound. It is known that greedy strategy can yield $O(V_f)$ regret bound when P_T is unknown [12]. Our algorithm can achieve these regret bound simultaneously due to the implicit update and novel learning parameters design.

Comparision with R-OBD and IOMD. Here we highlight the differences between IOMD-SOCO and algorithms R-OBD and IOMD, which sheds light on the design intuition of our

algorithm. When f_t could be observed before making decision x_t, R-OBD [11] chooses a point that minimizes the weighted sum of the hitting, and switching cost, i.e.,

$$x_t = \arg\min_{x\in\chi} f_t(x) + \alpha D_\Phi(x, x_{t-1}). \tag{3}$$

However, R-OBD cannot be applied to the case when f_t is unknown before making a decision x_t (i.e., online learning setting). IOMD-SOCO is designed to address such case. At round t, IOMD-SOCO chooses x_t that minimizes the sum of the previous loss f_{t-1} and a regularizer whose coefficient is time-varying, i.e.,

$$x_t = \arg\min_{x\in\chi} f_{t-1}(x) + \lambda_{t-1} D_\Phi(x, x_{t-1}). \tag{4}$$

The intuition behind it is that if the loss functions change slowly (i,e., f_t is close to f_{t-1}), the chosen x_t can be seen as an approximation to the point chosen by R-OBD.

IOMD has been proposed in [5] to tackle classic OCO problems and achieves an optimal static regret bound of $O(\min\{V_f, \sqrt{T}\})$ given generally convex loss functions. It has the same form as (4) except for the learning rates. We design appropriate learning parameters $\{\lambda_t\}$ in (4) to incorporate the switching costs, i.e., IOMD-SOCO. We will see in the later that this learning parameters setting indeed turns out to be the central focus in technical analysis in our proposed algorithm, and the tuning of the time-varying learning parameters in (4) is also the key challenge in solving SOCO.

Novelty of our analysis. Note that IOMD-SOCO is an extended version of IOMD by setting appropriate time-varying learning parameter λ_t. This learning parameters design is non-trivial and motivated by the upper bound of universal dynamic regret incurred by IOMD for arbitrary parameter sequence $\{\lambda_t\}$, as shown in the next lemma.

Lemma 1: Under Assumption 1, assume that λ_t is increasing over time t, choose actions according to the update rule (4) then we have

$$\text{Regret}(u_{1:T}) \le R^2\lambda_T + R\sum_{t=2}^{T}\lambda_t||u_t - u_{t-1}|| + \sum_{t=1}^{T}\delta_t \le R^2\lambda_T + R\sum_{t=2}^{T}(\lambda_t - 1)||u_t - u_{t-1}|| + \sum_{t=1}^{T}\delta_t + R\sum_{t=2}^{T}||u_t - u_{t-1}||,$$

where $\delta_t = f_t(x_t) - f_t(x_{t+1}) - \lambda_t D_\Phi(x_{t+1}, x_t) + \frac{1}{2}||x_{t+1} - x_t||^2$. Lemma 1 serves as a building block for proving Theorem 1. Ideally, to minimize the regret in Lemma 1, we would like to have λ_T to be as close as possible to the sum of δ_t over time while ensuring the gap between λ_t and 1 is small (e.g., $O(1)$). To do this, we introduce a recurrence in the computation of λ_t, i.e., we set $\lambda_t - 1 \propto \sum_{i=1}^{t-1}\delta_i$. Based on the above analysis, we set learning parameter $\lambda_t = \frac{1}{\beta^2}\sum_{i=1}^{t-1}\delta_i + 1$ in (4), for a parameter β to be defined later, and we obtain the IOMD-SOCO. It turns out that IOMD-SOCO could achieve a best-of-both-worlds regret bound of order $O(\min\{V_f, \sqrt{T + TP_T}\})$ by choosing appropriate $\beta^2 = O(P_T)$.

The analysis of Lemma 1 is different from IOMD [5], as we consider the universal dynamic regret and takes into account the switching costs. For example, to fit into our algorithm and incorporate with switching costs, we take a novel decomposition of the universal dynamic regret and handle it as

$$\begin{aligned}\text{Regret}(u_{1:T}) \le &\sum_{t=1}^{T}(f_t(x_t) - f_t(x_{t+1}) - (\lambda_t - 1)D_\Phi(x_{t+1}, x_t)) \\ &+ \sum_{t=1}^{T}\lambda_t(D_\Phi(u_t, x_t) - D_\Phi(u_t, x_{t+1})) \\ &+ \sum_{t=1}^{T}(D_\Phi(x_{t+1}, x_t) - \frac{1}{2}||x_{t+1} - x_t||^2).\end{aligned}$$

We also develop a new technique to bound the incurred term $\sum_{t=1}^{T}\lambda_t(D_\Phi(u_t, x_t) - D_\Phi(u_t, x_{t+1}))$ as $\lambda_1 D_\Phi(u_1, x_1) + R\sum_{t=2}^{T}\lambda_t||u_t - u_{t-1}|| + \sum_{t=2}^{T}(\lambda_t - \lambda_{t-1})D_\Phi(u_{t-1}, x_t)$. See details in the Appendix A of our online report [1].

A. Lower Bound

In this subsection, we investigate the lower bound in terms of universal dynamic regret for SOCO with squared l_p norm switching costs, given general convex loss functions. We will prove that the regret bounds presented in Theorem 1 are order-optimal, i.e., they match the lower bounds w.r.t the regularities P_T and V_f , which are indicated by the following two theorems, respectively.

Theorem 2: Let χ be the decision set with diameter R, for any online deterministic algorithm A on χ, there always exists sequences of convex functions $\{f_t\}_{t=1}^T$ and comparators $\{u_t\}_{t=1}^T$ such that

$$\text{Regret}(u_{1:T}) = \Omega(V_f). \tag{5}$$

Theorem 3: Let χ be the decision set with diameter R, for any online deterministic algorithm A on χ, there always exists sequences of convex functions $\{f_t\}_{t=1}^T$ and comparators $\{u_t\}_{t=1}^T$ such that

$$\text{Regret}(u_{1:T}) = \Omega(\sqrt{T + TP_T}). \tag{6}$$

Therefore, the regret bound in Theorem 1 cannot be improved in general, which demonstrates the optimality of our algorithm and the obtained dynamic regret bound.

Remark 2: When the switching cost is defined as squared l_2 norm, [30] proved an $O(\sqrt{T + TV_x})$ lower bound for restricted dynamic regret, where $\{x_t^*\}_{t=1}^T = \arg\min_{x_1,\ldots,x_T\in\chi}\sum_{t=1}^{T}(f_t(x_t) + c(x_t, x_{t-1}))$ and V_x is the path-length of sequence $\{x_t^*\}_{t=1}^T$. Their result is the special case of Theorem 3. To prove Theorem 3, we divide T into multiple phases and construct a special instance such that the comparator points remain unchanged in each phase. Then we relate the incurred dynamic regret with the minimax static regret to obtain the final result. To the best of our knowledge, the results in Theorems 2 and 3 are the first lower bounds for SOCO in terms of universal dynamic regret.

III. Extension: Parameter-free Setting

In Section II, we explored the analytical performance of IOMD-SOCO in terms of universal dynamic regret. We notice that the optimal tuning of λ_t or β in IOMD-SOCO requires the prior information of path-length P_T. Although it is a common assumption in previous work (e.g., [8], [10]–[12], [30]) to have

Algorithm 2 Hedge-IOMD-SOCO

1: **Initialize:** step size α, set $\mathcal{H}$.
2: Activate a set of experts $i = 1, 2, ..., |\mathcal{H}|$ by invoking IOMD-SOCO for each parameter $\beta_i \in \mathcal{H}$.
3: **for** round t = 1...T **do**
4: Receive x_t^i from each expert i
5: Choose the action $x_t = \sum_i w_t^i x_t^i$
6: Observe f_t, and send it to each expert i
7: Update the weight of each expert $i \in \{i = 1, 2, ..., |\mathcal{H}|\}$:
8: $w_{t+1}^i = \frac{w_t^i e^{-\alpha(\langle \nabla f_t(x_t), x_t^i - x_t\rangle + \frac{1}{2}||x_t^i - x_{t-1}^i||^2)}}{\sum_{j=1,...,|\mathcal{H}|} w_t^j e^{-\alpha(\langle \nabla f_t(x_t), x_t^j - x_t\rangle + \frac{1}{2}||x_t^j - x_{t-1}^j||^2)}}$
9: **end for**

prior knowledge of regularities (e.g., P_T or V_f), it is not true in many practical applications of SOCO. For example, P_T is unavailable ahead of time when the comparator sequence is the sequence of local minimizers of online loss functions (i.e., $\{\{\theta_t\}_{t=1}^T | \theta_t = \arg\min_{x\in\chi} f_t(x)\}$), since θ_t is unknown to the agent before making decision x_t. Hence, we analyze the performance guarantee of IOMD-SOCO when P_T is unknown in advance, which is shown in the following theorem.

Theorem 4: Under Assumption 1, when P_T is unknown in advance, we let $\beta^2 = R^2$, then IOMD-SOCO ensures that

$$\begin{aligned}\text{Regret}(u_{1:T}) &\leq R^2\lambda_T + R\sum_{t=2}^{T}\lambda_t||u_t - u_{t-1}|| + \sum_{t=1}^{T}\delta_t \\ &\leq O(\min\{P_T V_f, \sqrt{T}P_T\}).\end{aligned} \tag{7}$$

However, Theorem 4 shows that IOMD-SOCO with parameter-free learning rates has a great performance degradation than IOMD-SOCO with optimal learning rates ($\beta^2 = R^2 + RR_T$). Moreover, the regret bound in Theorem 4 may not be sublinear even when the regularities measure are sublinear, e.g., $P_T = O(\sqrt{T})$ or $P_T = V_f = O(\sqrt{T})$.

Thus, a natural problem is to design a parameter-free algorithm for SOCO. To the best of our knowledge, no prior algorithms achieve a parameter-free sublinear dynamic regret, even when the loss functions are known in advance. In this section, we will propose the parameter-free versions of IOMD-SOCO which gives a regret that is almost the same as the results in Theorem 1.

A. Tuning Parameters Using Ensemble Method

Inspired by the recent work [25], [27], we adopt the online ensemble method to tune the learning parameters when the regularities (e.g., P_T) are unknown. Next we give an explicit description of our resemble method, Hedge-IOMD-SOCO.

Hedge-IOMD-SOCO is a two-layer hierarchical structure. We run a set of experts at the lower level in parallel. At the higher level, we employ a meta-algorithm like Hedge [31] to track the best expert based on their performance, and output the final decision. Concretely, we initiate multiple expert-algorithms, each running IOMD-SOCO with a specific parameter β. At each round, experts send their decisions to the meta-algorithm, then experts receive a loss function from the meta-algorithm, and produce the decisions of the next round based on the incurred costs. The meta-algorithm first aggregates predictions from all the experts to make the final decision, i.e., the weighted average of received predictions (See line 5 in Algorithm 2). After observing the loss function, the meta-algorithm sends it to all experts and updates the weights of each expert according to the exponential-weighting scheme (See line 8 in Algorithm 2). To take into account the switching costs, different from previous ensemble methods [25], [27] for classic OCO, the loss of expert i we used in the Hedge is

$$f_t^i(x) = \langle \nabla f_t(x_t), x - x_t\rangle + \frac{1}{2}||x - x_{t-1}^i||^2,\ i = 1, 2, ..., |\mathcal{H}|.$$

As we can see, $f_t^i(x_t^i)$ incorporates the switching cost $\frac{1}{2}||x_t^i - x_{t-1}^i||^2$ of expert i to measure its performance. Next, we describe the design of the parameters for all experts.

Note that Theorem 1 shows the optimal parameter β is $\beta^* = \sqrt{R^2 + RP_T}$. The key idea is to construct a parameter pool $\mathcal{H}$ which contains β^* and search over $\mathcal{H}$ to identify it. Specifically, since $R \leq \beta^* = \sqrt{R^2 + RP_T} \leq \sqrt{R^2 + R^2T} = R\sqrt{T+1}$, we specify $\mathcal{H}$ as follows,

$$\mathcal{H} = \{\beta_i = R2^{i-1} | i \in [N], N = \lceil \frac{1}{2}\log(1+T)\rceil + 1\},$$

and each expert i initializes the parameter β_i. The reason we construct such a parameter pool is to ensure that there must exist an expert i, whose parameter β_i satisfies $\frac{\beta^*}{2} \leq \beta_i \leq \beta^*$. It turns out that in such case, our ensemble approach Hedge-IOMD-SOCO achieves an $O(\min\{V_f, \sqrt{T + TP_T}\})$ regret bound. We formally give this result in the following theorem.

Theorem 5: Under Assumption 1, let $\alpha = \frac{1}{\sqrt{T}}$, then Hedge-IOMD-SOCO ensures that

$$\text{Regret}(u_{1:T}) \leq O(\min\{\sqrt{T + TP_T}, \max\{V_f, \sqrt{T}\}\}). \tag{8}$$

Theorem 5 highlights that Hedge-IOMD-SOCO only incurs a slightly worse performance degradation (an additive $O(\sqrt{T})$ term) compared with IOMD-SOCO with optimal parameter $\beta^2 = R^2 + RR_T$ (Theorem 1). Without requiring any prior knowledge of P_T, the Hedge-IOMD-SOCO algorithm applies to broader scenarios than IOMD-SOCO. Moreover, it has a potential to recover an $O(\min\{V_f, \sqrt{T + TP_T}\})$ regret bound (e.g., $V_f \geq O(\sqrt{T})$) although P_T is unavailable ahead of time. Indeed, the regret guarantee achieved by Hedge-IOMD-SOCO is still less than $O(\sqrt{T + TP_T})$ and thus outperforms any of prior results listed in Table 1 (e.g., [11], [12], [30]) in terms of dynamic regret, whether they assumed the loss functions are known or not. One may argue that Hedge-IOMD-SOCO needs to initiate $N = \lceil \frac{1}{2}\log(1+T)\rceil + 1\rceil$ expert algorithms, in which the computation complexity maybe large when T is sufficiently large. In fact, the number of experts N we should initiated is less than 10 when the total round $T < 2^{18}$, in which the complexity is acceptable in most practical situations.

Proof sketch of Theorem 5. To prove Theorem 5, we cannot directly follow from the previous analysis of online resemble method [25], [27], [29]. As we mentioned before, the loss of each expert we used in the meta-algorithm takes into account the switching cost (Line 8 in Algorithm 2). This leads to significant changes in the proof. In our analysis, we first decompose the total regret as the regret of expert i

Algorithm 3 AdaIOMD-SOCO

Initialize: $i = 0,\ \lambda_1^0 = 1,\ Q_0 = \sqrt{2}R,\ C_0 = 0.$
for round $t = 1...T$ **do**
  Choose the action x_t, and then observe f_t
  $C_i = C_i + ||u_t - u_{t-1}||$
  if $C_i > Q_i$ **then**
    $i = i + 1$
    $Q_i = R2^i,\ \lambda_{t+1}^i = 1,\ C_i = 0,\ \beta_i^2 = R^2 + RQ_i$
    $x_{t+1} = x_t$
  else
    $x_{t+1} = \arg\min_{x\in\chi} f_t(x) + \lambda_{t-1}^i D_\Phi(x, x_t)$
    $\delta_t = f_t(x_t) - f_t(x_{t+1}) - (\lambda_t - 1)D_\Phi(x_{t+1}, x_t)$
    $\lambda_{t+1}^i = \lambda_t^i + \frac{1}{\beta_i^2}\delta_t$
  end if
end for

(expert-regret) plus the regret of Hedge with respect to expert i (meta-regret), i.e.,

$$\text{Regret}(u_{1:T}) = \underbrace{\sum_{t=1}^{T}(f_t(x_t) + \frac{1}{2}||x_t - x_{t-1}||^2) - \sum_{t=1}^{T}(f_t(x_t^i) + \frac{1}{2}||x_t^i - x_{t-1}^i||^2)}_{\text{meta-regret}} + \underbrace{\sum_{t=1}^{T}(f_t(x_t^i) + \frac{1}{2}||x_t^i - x_{t-1}^i||^2) - \sum_{t=1}^{T}(f_t(u_t) + \frac{1}{2}||u_t - u_{t-1}||^2)}_{\text{expert-regret}}.$$

The expert-regret can be bounded using Theorem 1. Note that our meta-regret involves the switching costs. To bound the meta-regret, we characterize the switching costs by term $||\boldsymbol{w}_{t+1} - \boldsymbol{w}_t||_1$, which is the l_1 distance of consecutive expert weight vector, i.e., we handle the meta-regret as: Meta-regret $\leq \sum_{t=1}^{T}(\sum_j w_t^j f_t^j(x_t^j) - f_t^i(x_t^i)) + \frac{1}{2}\sum_{t=1}^{T}(R^2||\boldsymbol{w}_t - \boldsymbol{w}_{t-1}||_1^2 + 2R^2||\boldsymbol{w}_t - \boldsymbol{w}_{t-1}||_1)$. Then we associate the update rule of $\boldsymbol{w}_t$ with Follow-the-regularized-leader (FTRL) method using entropic regularization. Since FTRL with entropic regularization could be seen as a general projection operation into a simplex, we use the fact that the projection into a simplex is Lipschitz-continuous to bound $||\boldsymbol{w}_{t+1} - \boldsymbol{w}_t||_1$. The proof details is deferred to the Appendix F of our online report [1].

B. Tuning Parameters Using the "Doubling Trick"

In some special scenarios P_T could be monitored and calculated on the fly. For example, when $u_t = \theta_t$ [30], we can observe the path length until current time t, i.e., $P_t = \sum_{\tau=1}^{t}||\theta_\tau - \theta_{\tau-1}||$. Another example is when $u_t = u$ [3], we directly have $P_T = O(1)$. In these special cases, we could use the strategy of the similar spirit to doubling trick to remove the dependence on P_T. Next, we present how to tune the learning parameter λ_t adaptively in IOMD-SOCO using this strategy when the value of P_T is unknown in advance.

Specifically, we run IOMD-SOCO in phases. At the beginning of each phase i, we start monitoring the path length. We denote C_i the accumulated path length at phase i. Once C_i reaches a certain threshold, we restart the IOMD-SOCO and double the threshold. In the algorithm design, the challenge is how to determine the condition of doubling the threshold. We introduce a quantity Q_i for each phase i to characterize the influence of accumulated implicit difference. When it is less than the accumulated observable path-length in phase i, we will restart the algorithm and double the threshold. To achieve the same regret bound as IOMD-SOCO, we let the update rule of learning parameter in phase i be $\lambda_{t+1}^i = \lambda_t^i + \delta_t/(R^2 + RQ_i)$. We illustrate this algorithmic approach in Algorithm 3, named AdaIOMD-SOCO. However, AdaIOMD-SOCO can only be used when P_T could be calculated on the fly. Next, we give the regret bound incurred by AdaIOMD-SOCO.

Theorem 6: Under Assumption 1, AdaIOMD-SOCO ensures that

$$\text{Regret}(u_{1:T}) \leq O(\min\{V_f, \sqrt{T + TP_T}\}). \tag{9}$$

The above theorem highlights that regret incurred by AdaIOMD-SOCO satisfies the same upper bound given in Theorem 1, and is better than the regret guarantee of online ensemble method. Thus, for comparator sequence whose path-length is unknown but could be calculated on the fly, AdaIOMD-SOCO is a better choice.

Remark 3: We stress that the idea of AdaIOMD-SOCO to remove the dependence on regularities is not the standard doubling trick (only with the similar spirit). Specifically, in order to have a fully adaptive learning rate, we tune it as a function of two quantities varying over time: accumulated path-length observed and accumulated δ_t incurred by the algorithm. While both quantities are increasing over time, they also appear both at the numerator and denominator of the learning rate λ_t. However, this would result in a non-monotone sequence of learning rates, thus contradicting the assumptions in Lemma 1. Also, we would like to point out that to the best of our knowledge there are no existing methods in the literature which tune the learning rates with non-monotone sequences.

IV. Applications

Here we show several real-world applications of SOCO including economic dispatch in power systems, trajectory tracking of moving bodies and cloud resource provisioning. We emphasize that none of these applications would be well-addressed without a parameter-free algorithm achieving sub-linear regret, which is not attainable by previous approaches.

Economic Dispatch in Power Systems. Consider a power network with N conventional generators and renewable energy supply [16]. At each time t, the dispatcher needs to decide the outputs of N generators, denoted as $x_t = [x_{t,1}, ..., x_{t,N}] \in \chi$, where χ is the set of feasible output vectors, and each generator i would incur $c_i(x_{t,i})$ generation cost. We let r_t be the renewable supply and d_t be the power demand at time t. The purpose of the dispatcher is to reduce the total generation cost while maintaining the power balance of supply and consumption: $\sum_{i=1}^{N} x_{t,i} + r_t = d_t$. To incorporate the imbalance penalty into the objective, we define the following cost function at each time t: $f_t(x_t) = \sum_{i=1}^{N} c_i(x_{t,i}) + \epsilon_t(\sum_{i=1}^{N} x_{t,i} - r_t - d_t)^2$, where ϵ_t is a penalty coefficient. Specifically, [20] modeled $c_i(x_{t,i})$ as a quadratic function w.r.t

$x_{t,i}$. Since the power demand d_t is revealed after the output, f_t can only be determined at the end of time t. In addition to the above costs, ramping process of conventional generators also incurs other costs such as maintenance and depreciation fee. These additional costs are called ramp costs and usually modeled as a quadratic function of the amount of ramp $\alpha \sum_{i=1}^{N} (x_{t,i} - x_{t-1,i})^2$ ([18], [23]), i.e., $\alpha||x_t - x_{t-1}||_2^2$. The economic dispatcher aims to minimize the incurred total costs over T rounds, i.e.,

$$\min_{\{x_t\}_{t=1}^T} \sum_{t=1}^{T} (f_t(x_t) + \alpha||x_t - x_{t-1}||_2^2).$$

Low-latency Video Streaming. In the scenario of content distribution networks (CDNs), edge servers are usually designed to accelerate the access of popular content (e.g., video streaming) by prefetching them in low-latency storage. This scenario can be simplified as a caching system with a remote server that contains a set of N unique files and a local cache of finite capacity C (also see [4]). Time is slotted such that at most k files are requested from users at a time. We denote $r_t \in \{0,1\}^N$ as the file requests vector at time t, where $||r_t||_1 \leq k$ and $r_t(i) = 1$ if and only if the i-th file is requested by users at time t. The cache can update the cached files at the beginning of each time slot. We denote $x_t \in [0,1]^N$ as the cache configuration at time t, where $||\boldsymbol{x}||_1 \leq C$ and $x_t(i)$ represents the fraction of file i cached in time t. As an example, in P2P data streaming and CDNs, large video files are composed of independently stored chunks. Let w_i be the obtained utility if file i is requested and hit, e.g., benefits due to the bandwidth saving from cache hits, or QoS improvement. Thus, the cache configuration x_t accrues in time t a utility as: $\sum_{i=1}^{N} w_i r_t(i) x_t(i)$. We also assume that changing the cache configuration from x_{t-1} to x_t incurs a switching cost as it needs to fetch files from the remote server. Literature (e.g., [19]) usually models it as the form of $\alpha||x_t - x_{t-1}||_1^2$. Putting everything together, the objective of cache is to maximize the overall obtained utility over a time horizon T:

$$\max_{\{x_t\}_{t=1}^T} \sum_{t=1}^{T} (\sum_{i=1}^{N} w_i r_t(i) x_t(i) - \alpha||x_t - x_{t-1}||_1^2).$$

Cloud Resource Allocation. Consider a could service provider that predicts and provides network, computing, and storage resources dynamically to meet the demands from applications [10]. At time t, we denote d_t as the demand vector where each dimension represents a type of resource requested, and x_t the vector of resources provisioned by the could service provider. The demand vector d_t is revealed after the decision x_t. To provide resource x_t, the provider incurs operational costs and switching costs. The operational costs typically include the monetary cost of retaining and using virtual machines, amortized capital costs, energy consumption, and delay cost when resources are under-provisioned. Previous literature (e.g., [9], [10]) usually models the operational costs in a general form of convex function $f(x_t; d_t)$ of the demands and the provisioned resources. Switching costs include the wear, tear and delay during server startup and shutdown, as well as the cost of virtual machine migration and data transfer. Following previous literature [10], the switching cost can be capture by the form of $\alpha||x_t - x_{t-1}||_2^2$, i.e., squared l_2 norm on provisioned resources change. Thus, the provider aims to minimize the total costs over a time horizon T:

$$\min_{\{x_t\}_{t=1}^T} \sum_{t=1}^{T} (f(x_t : d_t) + \alpha||x_t - x_{t-1}||_2^2).$$

Smoothed Online Regression. Consider the scenario of online regression [12], in which the learner wishes to fit a series of regularized regressors or classifiers to a time-varying data-set, without changing the estimators too much between rounds. Specifically, the learner aims to solve the following online optimization problem:

$$\min_{\{\theta_t\}_{t=1}^T \in R^{d \times T}} \sum_{t=1}^{T} f_t(\theta_t) + \frac{\lambda_1}{2}||\theta_t||_2^2 + \frac{\lambda_2}{2}||\theta_t - \theta_{t-1}||_2^2. \quad (10)$$

Here θ_t is the regressor at time t, f_t represents the regression loss at time t, and λ_1, λ_2 are the parameters of l_2 and smoothing regularizations, respectively. Our scenario includes many regression types such as Ridge Regression, i.e., $f_t(\theta_t) = ||X_t\theta_t - y_t||_2$, where $X_t \in R^{d \times n_t}$ is a data matrix, $y_t \in R^d$ is the response variable, and n_t is the number of samples at round t; and Logistic Regression, i.e., $f_t(\theta_t) = -\frac{1}{n_t} \sum_{i=1}^{n_t} \log(1 + e^{-y_{i,t}\theta_t^T x_{i,t}})$, where $x_{i,t} \in R^d$ is a feature vector, $y_{i,t} \in \{0,1\}$ is the corresponding binary label, and n_t is the number of samples in round t.

V. Numerical experiments

In this section, we conduct numerical experiments to validate the theoretical performance of our algorithms.

Experimental setting. We consider the smoothed online logistic regression problem as the numerical example, which is outlined in Section IV. Specifically, the objective is to solve the online optimization problem (10), where f_t is the logistic regression loss at time t. At time t, we generate $\{x_{i,t}\}_{i=1}^{n_t}$ as i.i.d. Gaussian vectors with mean $\mu_t \boldsymbol{I}$ and covariance $\sigma_t^2 \boldsymbol{I}$, and $\{y_{i,t}\}_{i=1}^{n_t}$ i.i.d. from Bernoulli(p_t). We let $d = 3$, $\chi = [-1,1]^3$, $n_t = 20$, and $\theta_0 = [0,0,0]$. We also set $\lambda_1 = \lambda_2 = 1$. As there are only a few works on the setting of generally convex loss functions without predictions, we choose OGD [10], [24] and OMD [30] initializations as our baselines and use the corresponding stepsizes in their results. We compare the time-averaged regrets of our algorithms with these baselines both on synthetic and real-world datasets.

Results for synthetic dataset. We simulate these methods on the synthetic dataset in which $V_f = o(\sqrt{T})$. To achieve this, we let $\mu_t = \mu$, where μ is sampled uniformly on $[-1,1]$, and $p_t = p$, where p is sampled uniformly on $(0,1)$. We also set $\sigma_t^2 = t^{-1}$. We first choose the best offline fixed regressor as the benchmark, i.e., $u_t = \theta^* = \arg\min_{\theta \in \chi} \sum_{t=1}^{T} f_t(\theta) + \frac{\lambda_1}{2}||\theta||_2^2, \forall t$. We can verify that in this case, $P_T = O(1)$ and we present the simulation results of IOMD-SOCO and baselines in Figure 1 (a). From this figure, we can see that IOMD-SOCO indeed guarantees a much small regret than $O(\sqrt{TP_T})$ when the accumulated variation of consecutive loss functions is small, which validates our theoretical result. The empirical performance of OGD and OMD in our experiment also coincides with their theoretical regret guarantee of the order $O(\sqrt{TP_T})$.

Then we choose the per-time optimal regressor sequence as the benchmark, i.e., $u_t = \arg\min_{\theta \in \chi} f_t(\theta) + \frac{\lambda_1}{2}||\theta||_2^2, \forall t$. Note that in this case P_T is unknown a priori and thus only methods Hedge-IOMD-SOCO and AdaIOMD-SOCP are

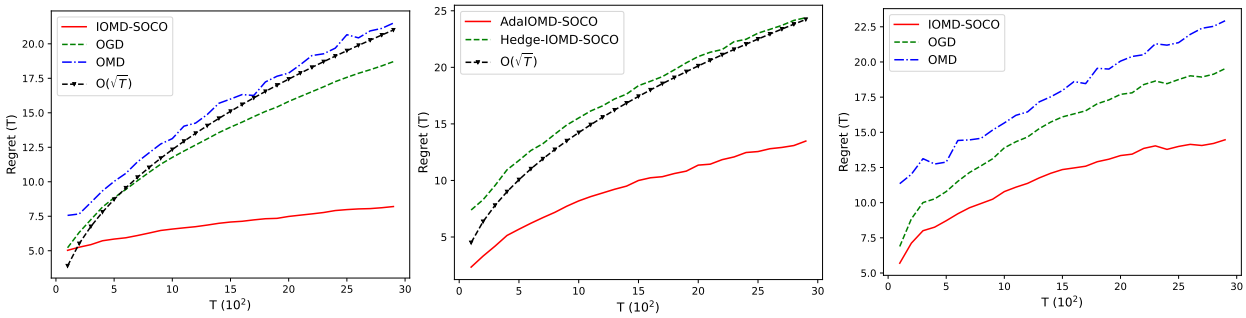

Fig. 1. (a) Results when $V_f = o(\sqrt{T})$ and $P_T = O(1)$; (b) Results when $V_f = o(\sqrt{T})$ and P_T is unknown; (c) Results for real data-set

applicable. We show their empirical results in Figure 1 (b). From this figure, we can see that Hedge-IOMD-SOCO and AdaIOMD-SOCP indeed achieve a regret less than $O(\sqrt{T})$ even the P_T is unknown a prior, which matches their theoretical results.

Results for real-world dataset. We also test these methods on the real dataset: USENET1 [13]. The distribution of data streams is time-varying for this dataset, which is the dynamic environment we considered. We still choose the best offline fixed regressor as the benchmark and compare the IOMD-SOCO and baselines in Figure 1 (c). Figure 1 (c) shows that IOMD-SOCO consistently produces the best performance when compared to the baseline methods.

REFERENCES

[1] Online report. https://cloud.tsinghua.edu.cn/f/5a25db215cea44139dd9/.

[2] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

[3] Aditya Bhaskara, Ashok Cutkosky, Ravi Kumar, and Manish Purohit. Power of hints for online learning with movement costs. In *International Conference on Artificial Intelligence and Statistics*, pages 2818–2826. PMLR, 2021.

[4] Rajarshi Bhattacharjee, Subhankar Banerjee, et al. Fundamental limits on the regret of online network-caching. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2):1–31, 2020.

[5] Nicolo Campolongo and Francesco Orabona. Temporal variability in implicit online learning. *arXiv preprint arXiv:2006.07503*, 2020.

[6] Niangjun Chen, Anish Agarwal, Adam Wierman, et al. Online convex optimization using predictions. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 191–204, 2015.

[7] Niangjun Chen, Joshua Comden, Zhenhua Liu, Anshul Gandhi, and Adam Wierman. Using predictions in online optimization: Looking forward with an eye on the past. *ACM SIGMETRICS Performance Evaluation Review*, 44(1):193–206, 2016.

[8] Niangjun Chen, Gautam Goel, and Adam Wierman. Smoothed online convex optimization in high dimensions via online balanced descent. In *Conference On Learning Theory*, pages 1574–1594. PMLR, 2018.

[9] Niangjun Chen, Xiaoqi Ren, Shaolei Ren, and Adam Wierman. Greening multi-tenant data center demand response. *Performance Evaluation*, 91:229–254, 2015.

[10] Joshua Comden, Sijie Yao, Niangjun Chen, Haipeng Xing, and Zhenhua Liu. Online optimization in cloud resource provisioning: Predictions, regrets, and algorithms. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(1):1–30, 2019.

[11] Gautam Goel, Yiheng Lin, Haoyuan Sun, and Adam Wierman. Beyond online balanced descent: An optimal algorithm for smoothed online optimization. *Advances in Neural Information Processing Systems*, 32:1875–1885, 2019.

[12] Gautam Goel and Adam Wierman. An online algorithm for smoothed regression and lqr control. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2504–2513. PMLR, 2019.

[13] Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. An ensemble of classifiers for coping with recurring contexts in data streams. In *ECAI 2008*, pages 763–764. IOS Press, 2008.

[14] Seung-Jun Kim and Geogios B Giannakis. An online convex optimization approach to real-time energy pricing for demand response. *IEEE Transactions on Smart Grid*, 8(6):2784–2793, 2016.

[15] Yingying Li and Na Li. Leveraging predictions in smoothed online convex optimization via gradient-based algorithms. *Advances in Neural Information Processing Systems*, 33:14520–14531, 2020.

[16] Yingying Li, Guannan Qu, and Na Li. Using predictions in online optimization with switching costs: A fast algorithm and a fundamental limit. In *2018 Annual American Control Conference (ACC)*, pages 3008–3013. IEEE, 2018.

[17] Yingying Li, Guannan Qu, and Na Li. Online optimization with predictions and switching costs: Fast algorithms and the fundamental limit. *IEEE Transactions on Automatic Control*, 2020.

[18] Reetabrata Mookherjee, Benjamin F Hobbs, Terry L Friesz, and Matthew A Rigdon. Dynamic oligopolistic competition on an electric power network with ramping costs and joint sales constraints. *Journal of Industrial & Management Optimization*, 4(3):425, 2008.

[19] Samrat Mukhopadhyay and Abhishek Sinha. Online caching with optimal switching regret. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 1546–1551. IEEE, 2021.

[20] Balakrishnan Narayanaswamy, Vikas K Garg, and TS Jayram. Online optimization for the smart (micro) grid. In *Proceedings of the 3rd international conference on future energy systems: where energy, computing and communication meet*, pages 1–10, 2012.

[21] Marcello Restelli, Edoardo Vittori, et al. Online gradient descent for online portfolio optimization with transaction costs. 2020.

[22] Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Online optimization with memory and competitive control. *arXiv preprint arXiv:2002.05318*, 2020.

[23] Makoto Tanaka. Real-time pricing with ramping costs: A new approach to managing a steep change in electricity demand. *Energy Policy*, 34(18):3634–3643, 2006.

[24] L. Zhang, W. Jiang, et al. Revisiting smoothed online learning. 2021.

[25] Lijun Zhang, Shiyin Lu, and Zhi-Hua Zhou. Adaptive online learning in dynamic environments. *arXiv preprint arXiv:1810.10815*, 2018.

[26] Lijun Zhang, Tianbao Yang, Jinfeng Yi, Rong Jin, and Zhi-Hua Zhou. Improved dynamic regret for non-degenerate functions. In *Advances in Neural Information Processing Systems*, pages 732–741, 2017.

[27] Yu-Jie Zhang, Peng Zhao, and Zhi-Hua Zhou. A simple online algorithm for competing with dynamic comparators. In *Conference on Uncertainty in Artificial Intelligence*, pages 390–399. PMLR, 2020.

[28] Peng Zhao and Lijun Zhang. Improved analysis for dynamic regret of strongly convex and smooth functions. *arXiv preprint arXiv:2006.05876*, 2020.

[29] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Dynamic regret of convex and smooth functions. *Advances in Neural Information Processing Systems*, 33, 2020.

[30] Yawei Zhao, Qian Zhao, Xingxing Zhang, En Zhu, Xinwang Liu, and Jianping Yin. Understand dynamic regret with switching cost for online decision making. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–21, 2020.

[31] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.