

Using Predictions in Online Optimization with Switching Costs: A Fast Algorithm and A Fundamental Limit

Yingying Li, Guannan Qu, and Na Li

Abstract—This paper studies an online optimization problem with switching costs and a finite prediction window. We propose a computationally efficient algorithm, Receding Horizon Gradient Descent (RHGD), which only requires a finite number of gradient evaluations at each time. We show that both the dynamic regret and the competitive ratio of the algorithm decay exponentially fast with the length of the prediction window. Moreover, we provide a fundamental lower bound on dynamic regret for general online algorithms with a finite prediction window. The lower bound matches the dynamic regret of our RHGD, meaning that the performance can not improve significantly even with more computation. Lastly, we present simulation results to test our algorithm numerically.

I. INTRODUCTION

A classic online convex optimization (OCO) problem considers a decision maker interacting with an uncertain and even adversarial environment for T stages. At each time $t \in \{1, \dots, T\}$, the decision maker picks an action x_t from a convex set X . Then the environment reveals a convex cost $f_t(\cdot)$. As a result, the decision maker suffers the cost $f_t(x_t)$ based on the chosen action. The goal is to minimize the total cost in T stages. Classic OCO has been studied for decades, with an focus on improving online algorithm performance measured by regrets and/or competitive ratio [1]–[4].

Recent years have witnessed a growing interest in applying online optimization to real-world systems, e.g. economic dispatch [5]–[7], data center scheduling [8], [9], electric vehical charging [10], [11], video streaming [12], and thermal control [13]. However, there are two features of these scenarios that are generally not captured by classic OCO literature: time coupling effect and prediction about the future.

Time coupling effect: While classic OCO setup assumes that stage costs $f_t(x_t)$ are completely decoupled with time, in reality it is usually not the case. For example, to change actions from x_{t-1} to x_t , decision makers usually suffer a switching cost or ramp cost $d(x_t - x_{t-1})$ [8], [9], [14], [15]. In this way, the stage cost becomes time coupled and is defined as: $C_t(x_{t-1}, x_t) := f_t(x_t) + d(x_t - x_{t-1})$.

Prediction: Classic OCO often models environment as adversary and assumes that no information is available about future cost functions. However, in most applications, there is certain amount of prediction about the future, especially the near future. For example, in electricity system, the system operator can obtain a good prediction about the future demand and generation [16] [17].

Recently, there are some studies from OCO community exploring the effect of prediction, but most of them do not consider time coupled stage costs [18], [19].

In contrast, there have been many control algorithms, in particular, Model Predictive Control (MPC, aka Receding Horizon Control), developed for decades to handle both the prediction effect and the time coupling effect. One major focus of MPC is to design control rules to stabilize a dynamical system. Additional goals include minimizing the total stage costs, as studied in economic MPC [20]–[22]. However, the classic MPC approaches require solving a large optimization problem at each time, which is usually computationally expensive. Though there have been many recent efforts trying to reduce the computational overhead, e.g. inexact MPC and suboptimal MPC [23]–[26], there are limited results on the efficiency loss of these algorithms, such as bounds on dynamic regret. This is partially due to the complexity of the underlying system dynamics. Lastly, other similar online control algorithms, such as Averaging Fixed Horizon Control (AFHC) [8], [9], [27], [28], suffer from the same problem of high computational costs.

Contribution of this paper: In this paper we consider an OCO problem with switching costs and a prediction window W . To design online algorithms for this problem, we first study the structure of offline gradient-based algorithms, and use it to motivate our online algorithm Receding Horizon Gradient Descent (RHGD). Our algorithm only requires $W + 1$ gradient evaluations at each time step, which is more computationally efficient compared to MPC or AFHC like algorithms. Besides, there is a smooth interpolation between our algorithm and classic online methods in settings without prediction or switching cost: when $W = 0$, our algorithm reduces to the classic online gradient descent.

In addition, we analyze the online performance of RHGD by comparing RHGD outputs and offline optimal solution. The comparison is measured by both dynamic regret and competitive ratio. We show that the dynamic regret of RHGD decays exponentially with W . Furthermore, under mild conditions on the costs, RHGD’s competitive ratio is upper bounded by $1 + O((1 + \frac{1}{Q_f})^W)$, also exponentially decaying with W . Moreover, we study the fundamental performance limit of online algorithms given a prediction window W . We give a lower bound on the dynamic regret which also decays exponentially with W . This surprisingly matches the asymptotic property of RHGD’s upper bound, meaning that even for online algorithms that use more computation, the performance can not be significantly better than RHGD. Lastly, we use the economic dispatch problem of electricity systems to numerically test our algorithm. The simulation is

This work was supported by NSF CAREER 1553407 and ARPA-E NODES. Y. Li, G. Qu and N. Li are with the School of Engineering and Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, USA (email: yingyingli@g.harvard.edu, gqu@g.harvard.edu, nali@seas.harvard.edu).

consistent with our theoretical results.

A. Notations

For vector $x \in \mathbb{R}^n$ and set $X \subset \mathbb{R}^n$, norm $\|x\|$ refers to the Euclidean norm, and $\Pi_X(x)$ denotes the projection of x onto set X . Besides, we denote the transpose of x as x' . The same applies to matrix transpose. In addition, X^T denotes the Cartesian product of T copies of set X . Moreover, we define $[T]$ as the set $\{1, \dots, T\}$ for a positive integer T . Finally, let $\nabla f(x)$ be the gradient of function $f(x)$.

II. PROBLEM FORMULATION

In this paper we consider a variation of online convex optimization where the algorithm is subject to an additional switching cost on the change of actions from one time step to the next. If nothing is known about the future, one can expect that no online algorithm can perform well. Fortunately, in many practical applications, prediction with high precision is indeed available for the near future, e.g. wind generation and load forecast [16] [17]. Thus in this paper, we consider a situation where the algorithm has information about the current and a limited number of future cost functions.

Formally, we consider online convex optimization over a finite time horizon $t \in [T] := \{1, \dots, T\}$. At each time step t , a sequence of costs f_t, \dots, f_{t+W-1} from a function class is revealed to the online algorithm. This W -lookahead window is perfect in the sense that these are the true cost functions the algorithm will experience in future time steps.¹ Given perfect W -lookahead, the decision maker needs to pick an action x_t from a compact convex action set $X \subset \mathbb{R}^n$. The goal is to minimize the total cost of online decisions $\{x_1, \dots, x_T\}$ in T stages. The total cost is given by

$$C_1^T(x_1, \dots, x_T) := \sum_{t=1}^T \left(f_t(x_t) + \frac{\beta}{2} \|x_t - x_{t-1}\|^2 \right) \quad (1)$$

where x_0 is given and $\beta \geq 0$ is a weight parameter. Without loss of generality, we assume $x_0 = 0$. Notice that here we consider quadratic switching cost functions, but the analysis can extend to other switching cost functions with properties such as monotonicity, convexity, and smoothness.

For the purpose of theoretical analysis, this paper considers a function class $\mathcal{F}_X(\alpha, l, G)$ which consists of functions defined on X that are α -strongly convex, differentiable and have l -Lipschitz continuous gradients which are bounded by G . Mathematically, this means for every $t \in [T]$ and any $x_t, y_t \in X$, the following inequalities hold,

- i) $f_t(y_t) \geq f_t(x_t) + \langle \nabla f_t(x_t), y_t - x_t \rangle + \frac{\alpha}{2} \|y_t - x_t\|^2$
- ii) $\|\nabla f_t(y_t) - \nabla f_t(x_t)\| \leq l \|y_t - x_t\|$
- iii) $\|\nabla f_t(x_t)\| \leq G$

Given the above property for each f_t , we can show the following nice properties hold for the total cost function $C_1^T(\cdot)$. The proof is deferred to Appendix A.

¹Although this assumption might be unrealistic, it is a good starting point to study the effect of prediction on online decision making. We will introduce prediction error in the future work.

Lemma 1. $C_1^T(x_1, \dots, x_T)$ is α -strongly convex and has L -Lipschitz continuous gradient, where $L = l + 4\beta$.

The problem setup has natural applications in many areas. Here we briefly discuss two applications.

Example 1. (*Economic Dispatch in Power Systems.*) Consider a power network with conventional generators and renewable energy supply. At time t , let $x_t = \{x_{t,i}\}_{i=1}^n$ be the outputs of n generators and X be the set of feasible outputs. The generation cost of generator i is $c^i(x_{t,i})$. The renewable supply is r_t and the power demand is d_t .

At time t , the goal of economic dispatch is to reduce total generation cost while maintaining power balance: $\sum_{i=1}^n x_{t,i} + r_t = d_t$. Thus we incorporate imbalance penalty into the objective and consider the cost function

$$f_t(x_t) = \sum_{i=1}^n c^i(x_{t,i}) + \xi_t \left(\sum_{i=1}^n x_{t,i} - r_t - d_t \right)^2$$

where ξ_t is a penalty factor. In literature, $c^i(x_{t,i})$ is usually modeled as a quadratic function within capacity limit [5]. Thus, $f_t(x_t)$ belongs to class $\mathcal{F}_X(\alpha, l, G)$.

In addition to the costs above, ramping process of conventional generators also incurs significant costs, e.g. maintenance and depreciation fee. In literature, such costs are referred as ramp costs and modeled as a quadratic function of ramping rate $\frac{\beta}{2} \|x_t - x_{t-1}\|^2 := \sum_{i=1}^n \frac{\beta}{2} \|x_{t,i} - x_{t-1,i}\|^2$ [14] [15]. As a result, the objective of economic dispatch for T stages is to minimize total costs including ramp costs

$$\min_{x_t} \sum_{t=1}^T \left(f_t(x_t) + \frac{\beta}{2} \|x_t - x_{t-1}\|^2 \right)$$

Although demand and renewable supply are random, prediction are available for a short time window [16] [17].

Example 2. (*Trajectory Tracking.*) Consider a simple dynamical system $x_{t+1} = x_t + u_t$, where x_t is the location of a robot, u_t is the control action (velocity of the robot). Let y_t be the location of the target at time t , and the tracking error is given by $f_t(x_t) = \frac{1}{2} \|x_t - y_t\|^2$. There will also be an energy loss for each control action, given by $\frac{\beta}{2} \|u_t\|^2 = \frac{\beta}{2} \|x_{t+1} - x_t\|^2$. The objective is to minimize the sum of the tracking error and the energy loss,

$$\min_{x_t} \sum_{t=0}^{T-1} \left(f_t(x_t) + \frac{\beta}{2} \|x_{t+1} - x_t\|^2 \right) + f_T(x_T).$$

In reality, there is usually a lookahead window W for the target trajectory y_t [29].

A. Performance Metrics

There are two popular ways to analyze an algorithm's online performance, *regret* and *competitive* analysis. In this paper, we will use dynamic regret and competitive ratio as our performance metrics.

Given a realization of cost functions f_1, \dots, f_T , the optimal offline solution is given by

$$\{x_1^*, \dots, x_T^*\} \in \arg \min_{x_t \in X, t \in [T]} C_1^T(x_1, \dots, x_T)$$

Let \mathcal{A} denote an online algorithm and $x_1^{\mathcal{A}}, \dots, x_T^{\mathcal{A}}$ be the action taken by this algorithm. Dynamic regret compares the costs of algorithm \mathcal{A} to the optimal offline cost,

$$\text{Reg}(\mathcal{A}) = C_1^T(x_1^{\mathcal{A}}, \dots, x_T^{\mathcal{A}}) - C_1^T(x_1^*, \dots, x_T^*).$$

Ideally, it is desirable to have regret grow sublinearly with T , because it means that the algorithm has no regret on average as $T \rightarrow \infty$. Unfortunately, dynamic regret depends on how fast the cost changes and is usually impossible to be sublinear in T given high fluctuation in costs (see our Section V and [30]).² Therefore, in literature, the dynamic regret is usually characterized by the environment/cost dynamics measures, such as path length. For instance, path length can be defined as $\sum_{t=1}^T \|y_{t+1}^* - y_t^*\|$ where $y_t^* \in \arg \min_{x_t \in X} f_t(x_t)$. An overview on different definitions of path lengths can be found in [2]. Most literature, as well as this paper, looks for an algorithm that guarantees sublinear regret when path length is sublinear in T [2], [4], [19], [30].

The other metric used in this paper is competitive ratio, which is the ratio between algorithm performance and the optimal offline cost

$$\text{CR}(\mathcal{A}) = \frac{C_1^T(x_1^{\mathcal{A}}, \dots, x_T^{\mathcal{A}})}{C_1^T(x_1^*, \dots, x_T^*)}.$$

To make this definition useful, an underlying assumption is that the cost function $f_t(x_t)$ is positive and not close to 0 [8]. An online algorithm is considered to be successful if $\text{CR}(\mathcal{A})$ can be upper bounded by a constant value.

III. RECEDING HORIZON GRADIENT DESCENT

In this section, we will introduce our online algorithm, Receding Horizon Gradient Descent (RHGD), which uses prediction and only needs a finite number of gradient updates. Before presenting the algorithm we first discuss an offline gradient algorithm, whose structure motivates our design of RHGD and helps us analyze its online performance.

A. Offline Optimization

Given a realization of cost functions f_1, \dots, f_T , the offline optimization is given by

$$\begin{aligned} \min C_1^T(x_1, \dots, x_T) \\ \text{s.t. } x_t \in X, \quad t \in [T] \end{aligned} \quad (2)$$

One simple algorithm to solve this problem is projected gradient descent (GD). Denote the k th update of x_t by GD as $x_t^{(k)}$. Given initial points $x_1^{(0)}, \dots, x_T^{(0)}$, the updating rule of GD is given by

$$x_t^{(k+1)} = \Pi_X \left(x_t^{(k)} - \eta g_t(x_1^{(k)}, \dots, x_T^{(k)}) \right) \quad (3)$$

²If the performance metric is static regret, i.e. difference between the online cost and the minimal cost over a static action, $C_1^T(x_1^{\mathcal{A}}, \dots, x_T^{\mathcal{A}}) - \min_{x \in X} C_1^T(x, \dots, x)$, there exist algorithms that achieve sublinear static regrets. However, static regret is not a suitable metric for the online problems we consider in this paper because the optimal action should be time-varying rather than being static.

where $g_t(x_1, \dots, x_T)$ denotes the partial gradient of $C_1^T(x_1, \dots, x_T)$ with respect to x_t , i.e.

$$g_t(x_1, \dots, x_T) := \left. \frac{\partial C_1^T}{\partial x_t} \right|_{(x_1, \dots, x_T)}.$$

Thanks to the special structure of our problem, $g_t(\cdot)$ only depends on x_t and its neighboring actions x_{t-1} and x_{t+1} for $t < T$. When $t = T$, it only depends on x_T and x_{T-1} . The explicit forms of gradients are given by

$$\begin{aligned} g_t(\cdot) &= \nabla f_t(x_t) + \beta(x_t - x_{t-1}) + \beta(x_t - x_{t+1}), \quad t < T \\ g_T(\cdot) &= \nabla f_T(x_T) + \beta(x_T - x_{T-1}) \end{aligned}$$

For ease of exposition, we abuse the notation by denoting $g_t(x_1, \dots, x_T)$ as $g_t(x_{t-1}, x_t, x_{t+1})$ since g_t only depends on x_{t-1}, x_t, x_{t+1} . Accordingly, (3) is rewritten as

$$x_t^{(k+1)} = \Pi_X \left(x_t^{(k)} - \eta g_t(x_{t-1}^{(k)}, x_t^{(k)}, x_{t+1}^{(k)}) \right) \quad (4)$$

This structure motivates our design of RHGD below.

B. Our Online Algorithm

Algorithm 1: Receding Horizon Gradient Descent (RHGD)

Input: x_0 , prediction window length W , stepsize γ, η

Initialize: Let x_s^t denote the value of x_s at time t where $s = 1, \dots, T$ and $t = s - W, s - W + 1, \dots, s$. Set $x_1^{1-W} = x_0$.

Update:

For $t = 2 - W$ to T

I) Initialize value for x_{t+W} .

If $t + W \leq T$,

$$x_{t+W}^t = \Pi_X \left(x_{t+W-1}^{t-1} - \gamma \nabla f_{t+W-1}(x_{t+W-1}^{t-1}) \right) \quad (5)$$

end

II) Update value for x_t, \dots, x_{t+W-1} backwards.

For $s = \min(t + W - 1, T) : -1 : \max(t, 1)$,

$$x_s^t = \Pi_X \left(x_s^{t-1} - \eta g_s(x_{s-1}^{t-2}, x_s^{t-1}, x_{s+1}^t) \right) \quad (6)$$

end

end

Output: $x_t^t \rightarrow x_t^{\text{RHGD}}$ at time t .

Due to the special structure of GD as in (4), we are able to implement the algorithm (4) in an online fashion up to $k = W$ updates. The algorithm is formally provided in Algorithm 1 and named Receding Horizon Gradient Descent (RHGD). Here, we explain the logic and meaning behind our algorithm and discuss the connection between RHGD and the offline GD algorithm as in (4).

We start with explaining the notations used in Algorithm 1. Firstly, x_s^t denotes the value of x_s at time t where $s \in \{1, \dots, T\}$ and $t \in \{s - W, s - W + 1, \dots, s\}$.³ This means that for each x_s , the first time it is assigned a value is at

³To ease notation, here we augment the time steps by including nonpositive values, i.e. allowing $t = 1 - W, \dots, 0$.

time $t = s - W$. Thus x_s^{s-W} is called the initial value of x_s using offline optimization language. Since at $t = s - W$, the cost function of f_s is still unknown, so x_s^{s-W} is obtained using the following rule

$$x_s^{s-W} = \Pi_X \left(x_{s-1}^{s-1-W} - \gamma \nabla f_{s-1}(x_{s-1}^{s-1-W}) \right).$$

where x_{s-1}^{s-1-W} stands for the initial value of x_{s-1} , and we use x_0 as initials of x_1 . In fact, this rule is (5) in Algorithm 1 by plugging in $t = s - W$. Moreover, this initialization rule is inspired by the classic Online Gradient Descent (OGD), as discussed in Remark 1 at the end of this section.

Starting from time $s - W + 1$, $f_s(\cdot)$ becomes available and its gradient $\nabla f_s(\cdot)$ is used to update x_s . At each time $t \in \{s - W + 1, s - W + 2, \dots, s\}$, RHGD updates x_s for once, so x_s^t is the value of x_s after $k = t - (s - W)$ updates for $k \in \{1, \dots, W\}$. The updating rule is in fact the same as GD (4), which is given in (6) and copied below

$$x_s^t = \Pi_X \left(x_s^{t-1} - \eta g_s(x_s^{t-2}, x_s^{t-1}, x_s^{t+1}) \right)$$

Although the arguments of $g_s(\cdot)$ here do not share the same superscript, they share the same k value, i.e. $t - 2 - (s - 1 - W) = t - 1 - (s - W) = t - (s - W) := k$, meaning that all of them are values after k gradient updates. As a result, the online updating rule (6) is the same as the offline GD (4), until $t = s$ or after $k = s - (s - W) = W$ updates. At time s , the W th updated value x_s^s is chosen to be RHGD's final decision, denoted as x_s^{RHGD} .

To sum up, at each time $t \in \{2 - W, \dots, T\}$, RHGD

- i) initializes the value for x_{t+W} according to (5),
- ii) updates the values for x_t, \dots, x_{t+W-1} backwards according to (6)⁴,

and this gives Algorithm 1.

The connection between RHGD and offline GD is crucial to our analytical study in the next section. Thus we make a formal statement on their relationship below.

Lemma 2. *Given the same stepsize η , if the offline GD's initial points satisfy*

$$x_s^{(0)} = x_s^{s-W}, \forall s \in [T]$$

then the output of RHGD is the same as offline GD after W iterations:

$$x_s^{(W)} = x_s^s, \forall s \in [T].$$

Proof. The main idea of the proof has already been discussed above. We omit the details due to space limit. \square

Remark 1. (Connection with Online Gradient Descent (OGD)) *Notice that the initialization rule (5) is the same as classic Online Gradient Descent (OGD) in online optimization without switching cost [1]:*

$$x_s = \Pi_X \left(x_{s-1} - \gamma \nabla f_{s-1}(x_{s-1}) \right), \forall s \in [T]. \quad (7)$$

Therefore, initial values x_s^{s-W} are outputs of OGD. When $W = 0$, our RHGD reduces to OGD.

⁴Backward updating makes x_{s+1}^t available for calculating x_s^t by (6).

IV. PERFORMANCE GUARANTEE

In this section, we analyze the performance of RHGD by comparing with the optimal offline solution. We provide an upper bound on dynamic regret which decays exponentially with W . In addition, we show that RHGD's competitive ratio is bounded by $1 + O\left(\left(1 - \frac{1}{Q_f}\right)^W\right)$ under mild condition.

A. Dynamic Regret

Here we give an upper bound on RHGD's dynamic regret.

Theorem 1. *Given $f_t(\cdot) \in \mathcal{F}_X(\alpha, l, G)$ for $t \in [T]$, and stepsizes $\gamma = 1/l$, $\eta = 1/L$, the dynamic regret of RHGD is upper bounded by*

$$\text{Reg}(RHGD) \leq \delta \left(1 - \frac{1}{Q_f}\right)^W \sum_{t=1}^T \|y_t^* - y_{t-1}^*\| \quad (8)$$

where $\delta = (\beta/l + 1) \frac{GQ_f}{(1-\kappa)}$, $\kappa = \sqrt{1 - \frac{\alpha}{l}}$, $Q_f = \frac{L}{\alpha}$, $y_t^* = \arg \min_X f_t(x_t)$, and $\sum_t \|y_t^* - y_{t-1}^*\|$ is known as path length.

Before the proof, we make a few comments on the bound.

Firstly, notice that the bound in (8) depends on cost variation measure, i.e. $\sum_{t=1}^T \|y_t^* - y_{t-1}^*\|$, which is also known as path length and commonly used in dynamic regret's upper bound in literature [2]. In fact, in section V we will show that this dependence is inevitable: given a finite lookahead window W , any online algorithm's dynamic regret is lower bounded by the path length up to a factor.

Secondly, the upper bound decays exponentially fast with the prediction window W . Thus, RHGD's performance improves significantly by increasing the lookahead window. This means that our algorithm uses the prediction information effectively.

Now we prove the theorem. The main idea is to use Lemma 2 and the convergence rate of the offline GD algorithm to bound the dynamic regret. In addition, since the initial values of RHGD are generated by OGD (5), the initial values' dynamic regret can also be bounded.

Although OGD is well studied [1], [2], to our best knowledge, the upper bound of OGD's dynamic regret for online optimization with switching cost has not been explored. Thus we present it here.

Lemma 3. *Given $f_t(\cdot) \in \mathcal{F}_X(\alpha, l, G)$ for $t \in [T]$, and stepsize $\gamma = 1/l$, the dynamic regret of OGD satisfies*

$$\text{Reg}(OGD) \leq (\beta/l + 1) \frac{G}{(1-\kappa)} \sum_{t=1}^T \|y_t^* - y_{t-1}^*\| \quad (9)$$

where $\kappa = \sqrt{1 - \frac{\alpha}{l}}$ and $y_t^* = \arg \min_X f_t(x_t)$. $\sum_{t=1}^T \|y_t^* - y_{t-1}^*\|$ is known as path length.

Proof. See Appendix B. \square

Now we are ready to prove Theorem 1.

Proof of Theorem 1: Applying Lemma 2, we can convert the dynamic regret of RHGD to the objective error of offline GD after W iterations

$$\text{Reg}(RHGD) = C_1^T(x_1^{(W)}, \dots, x_T^{(W)}) - C_1^T(x_1^*, \dots, x_T^*)$$

According to convergence rate of offline gradient descent for strongly convex and smooth functions, we have

$$\begin{aligned} & \text{Reg}(RHGD) \\ & \leq (C_1^T(x_1^{(0)}, \dots, x_T^{(0)}) - C_1^T(x_1^*, \dots, x_T^*))Q_f(1 - \frac{1}{Q_f})^W \end{aligned}$$

From Theorem 2, we know that initial values of offline GD is the same as the initial values of RHGD, which is also the outputs of OGD. As a result,

$$\text{Reg}(OGD) = C_1^T(x_1^{(0)}, \dots, x_T^{(0)}) - C_1^T(x_1^*, \dots, x_T^*)$$

Thus, by applying Lemma 3 we have the upper bound. \square

B. Upper Bound on Competitive Ratio

The following will show RHGD has constant competitive ratio under a mild assumption on stage costs.

Remember that competitive ratio only makes sense when $f_t(x_t)$ is positive and not close to zero. Specifically, we impose the following assumption as used in literature [8].

Assumption 1. For any t , there exists $e > 0$, such that $f_t(x_t) \geq e\|x_t\|$.

This assumption usually holds when cost $f_t(x_t)$ includes electricity charge or raw material cost with x_t representing amount of output or the number of device [14] [8].

Given Assumption 1, RHGD's competitive ratio can be bounded by a constant, which exponentially decays with W .

Theorem 2. Given Assumption 1, we have

$$\text{CR}(RHGD) \leq 1 + \frac{2\delta}{e}(1 - \frac{1}{Q_f})^W$$

where $\delta = (\beta/l + 1)\frac{G}{(1-\kappa)}$, $\kappa = \sqrt{1 - \frac{\alpha}{l}}$, $Q_f = \frac{l}{\alpha}$ and e is defined in Assumption 1.

Proof. When $\sum_{t=1}^T \|y_t^*\| \neq 0$,

$$\begin{aligned} \text{CR}(RHGD) & \leq 1 + \delta(1 - \frac{1}{Q_f})^W \frac{\sum_{t=1}^T \|y_t^* - y_{t-1}^*\|}{C_1^T(x_1^*, \dots, x_T^*)} \\ & \leq 1 + \delta(1 - \frac{1}{Q_f})^W \frac{\sum_{t=1}^T \|y_t^* - y_{t-1}^*\|}{\sum_{t=1}^T f_t(x_t^*)} \\ & \leq 1 + \delta(1 - \frac{1}{Q_f})^W \frac{\sum_{t=1}^T \|y_t^* - y_{t-1}^*\|}{\sum_{t=1}^T f_t(y_t^*)} \\ & \leq 1 + \delta(1 - \frac{1}{Q_f})^W \frac{\sum_{t=1}^T 2\|y_t^*\|}{\sum_{t=1}^T e\|y_t^*\|} \\ & = 1 + \frac{2\delta}{e}(1 - \frac{1}{Q_f})^W \end{aligned}$$

The first inequality is from Theorem 1. The second one is by throwing away switching cost. The Third one is because y_t^* minimizes $f_t(\cdot)$. The last inequality is due to triangle inequality of norm, and Assumption 1.

When $\sum_{t=1}^T \|y_t^*\| = 0$, then competitive ratio is 1, which also satisfies this upper bound. \square

V. LOWER BOUND

This section studies the performance limit of online algorithms given perfect prediction window W . For any online algorithm, we show that i) the dynamic regret decays at most exponentially fast with respect to the prediction window W ; ii) it is impossible to achieve sublinear regret in our setting given constant W and drastic fluctuations of costs, which aligns with results for other settings [3], [30].

We study the lower bound of online algorithm performance by constructing a special type of cost functions and introducing a characterization on all online algorithms.

Specifically, we consider costs $f_t(x_t) = \frac{\alpha}{2}(x_t - y_t^*)^2$, constraint $X = [-1, 1]$ and unknown parameters $y_t^* \in X$. Note that $y_t^* = \arg \min_X f_t(x_t)$. It is evident that $f_t(x_t) \in \mathcal{F}_X(\alpha, l, G)$ for $l = \alpha$ and $G = 2\alpha$. Accordingly, the Lipschitz factor of total cost is $L = \alpha + 4\beta$, and $Q_f = \frac{l}{\alpha}$.

Any online algorithm with perfect prediction window W can be captured by a series of maps $\{\mathcal{A}_t\}_{t=1}^T$. Each \mathcal{A}_t maps the information that is available at time t to an action in X

$$x_t = \mathcal{A}_t(y_1^*, y_2^*, \dots, y_{\min(t+W-1, T)}^*). \quad (10)$$

Notice that algorithms in the type of (10) include essentially every possible deterministic online algorithm, because the only limitation we impose on (10) is that the decision maker does not know y_{t+W}^*, \dots, y_T^* .

By showing a lower bound of dynamic regret of all online algorithms (10) for $f_t(\cdot)$ defined above, we are able to reveal the fundamental limit of online algorithm performance. A formal statement is given in Theorem 3.

Theorem 3. Define $\rho = \frac{\sqrt{Q_f}-1}{\sqrt{Q_f}+1}$. When $T \geq \max(\log_{\rho} \frac{\sqrt{1-\rho^2}}{2}, 2W)$, for any online algorithm \mathcal{A} in the form of (10), there always exists some y_1^*, \dots, y_T^* , such that the dynamic regret of the algorithm is lower bounded by

$$\text{Reg}(\mathcal{A}) \geq \frac{\alpha^2(1-\rho^2)^2}{32L} (\frac{\sqrt{Q_f}-1}{\sqrt{Q_f}+1})^{2W} \sum_{t=1}^T \|y_t^* - y_{t-1}^*\|.$$

Proof. Proof is by construction, which has been discussed above. Technical details can be found in Appendix C. \square

Remark 2. For any online algorithm, Theorem 3 shows that the dynamic regret decays at most exponentially fast with respect to the prediction window W . On the other hand, our algorithm RHGD also achieves a dynamic regret that at least decays exponentially with W , as stated in Theorem 1. It should be noted that the decaying rates are different. The lower bound decaying rate is $\frac{\sqrt{Q_f}-1}{\sqrt{Q_f}+1} \in [1 - \frac{2}{\sqrt{Q_f}}, 1 - \frac{1}{\sqrt{Q_f}}]$, while the decaying rate of our algorithm is $1 - \frac{1}{Q_f}$, which is slower than that of the lower bound for large Q_f . This gap is expected since algorithm (10) does not place any restriction on the computation or storage needs of the algorithm, while our algorithm RHGD only uses finite memory and $W + 1$ gradient evaluations at each time step. However, we expect that this gap can be bridged by using Nesterov Acceleration techniques in crafting our online gradient algorithms. We will leave this as our future work.

Remark 3. Notice that the lower bound in Theorem 3 depends on path length, $\sum_{t=1}^T \|y_t^* - y_{t-1}^*\|$, which captures the fluctuations of cost function $f_t(\cdot)$. Thus when path length is $O(T)$, Theorem 3 shows a $O(T)$ dynamic regret given constant W . Similar results have already been discovered in other settings [3], [30].

VI. A NUMERICAL STUDY: ECONOMIC DISPATCH

This section presents a numerical experiment of RHGD, an economic dispatch problem as introduced in Example 1.

In the simulation, we consider three conventional generators with quadratic costs given below.

$$\begin{aligned} c^1(x_{t,1}) &= (x_{t,1})^2 + 15x_{t,1} + 10 \\ c^2(x_{t,2}) &= 1.2(x_{t,2})^2 + 10x_{t,2} + 27 \\ c^3(x_{t,3}) &= 1.4(x_{t,3})^2 + 6x_{t,3} + 21 \end{aligned}$$

Besides, we consider a high-penetration of wind supply as shown in Figure 1 (b) where the data is from [31]. Figure 1 (a) depicts the load profile of New Hampshire from June 9 to June 12 in 2017 [32]. For simplicity, we let $\xi_t = \xi = 1.2$ and $\beta = 1$. In the simulation, each t corresponds to an hour.

Figure 1 (c) and (d) respectively present the dynamic regret and the competitive ratio of RHGD in log scale as a function of prediction window W . Notice that when $W = 0$, i.e. without prediction, RHGD reduces to classic OGD. When W increase from 0 to 4, both dynamic regret and competitive ratio enjoy a dramatic decrease. This shows that even with a small prediction window, RHGD can improve the online decision to a large degree. When $W > 4$, Figure 1 (c) and (d) show a linear decrease in log scale. This demonstrates that both dynamic regret and competitive ratio of RHGD decay exponentially with W .

Moreover, we compare RHGD with another fast online algorithm in literature: suboptimal MPC. Specifically, we apply the algorithm in [26]. The terminal cost is chosen to be $\frac{\beta}{2} \|x_{t+1} - x_t\|^2$ and the initial values follows OGD. To have a fair comparison, we evaluate the same number of gradients at each time for both algorithms. We also plot the dynamic regret and the competitive ratio of the suboptimal MPC in Figure 1 (c) and (d) respective. Interestingly, this algorithm and our RHGD share similar performance, with RHGD performs slightly better than suboptimal MPC when W is large. One interesting future direction is to formally analyze the regret and competitive ratio of suboptimal MPC.

VII. CONCLUSION

In this paper we propose a computational efficient online algorithm, RHGD, for online convex optimization problems with ramping cost. The algorithm uses W steps of prediction and only needs $W + 1$ steps of gradient evaluation at each time step. We show that the dynamic regret and the competitive ratio of RHGD decay exponentially fast with the prediction window W . This decaying rate matches a lower bound of general online algorithms, meaning that more computation can not achieves much better performance than our RHGD. There are many interesting future directions, such as i) generalizing the method to handle imperfect

prediction, ii) bridging the gap between the performance of RHGD and the lower bound by using Nesterov acceleration methods, iii) designing and studying other computational efficient online algorithms such as suboptimal MPC.

REFERENCES

- [1] E. Hazan, *Introduction to Online Convex Optimization*, ser. Foundations and Trends(r) in Optimization Series. Now Publishers, 2016. [Online]. Available: <https://books.google.com/books?id=IFxLvGAAAJ>
- [2] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 7195–7201.
- [3] L. Andrew, S. Barman, K. Ligett, M. Lin, A. Meyerson, A. Roytman, and A. Wierman, "A tale of two metrics: Simultaneous bounds on competitiveness and regret," in *Conference on Learning Theory*, 2013, pp. 741–763.
- [4] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.
- [5] B. Narayanaswamy, V. K. Garg, and T. 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*. ACM, 2012, p. 19.
- [6] M. Moeini-Aghtaie, P. Dehghanian, M. Fotuhi-Firuzabad, and A. Abaspour, "Multiagent genetic algorithm: an online probabilistic view on economic dispatch of energy hubs constrained by wind availability," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 2, pp. 699–708, 2014.
- [7] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Transactions on Signal Processing*, 2017.
- [8] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew, "Online algorithms for geographical load balancing," in *Green Computing Conference (IGCC), 2012 International*. IEEE, 2012, pp. 1–10.
- [9] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 5, pp. 1378–1391, 2013.
- [10] L. Gan, A. Wierman, U. Topcu, N. Chen, and S. H. Low, "Real-time deferrable load control: handling the uncertainties of renewable generation," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 3, pp. 77–79, 2014.
- [11] S.-J. Kim and G. B. Giannakis, "Real-time electricity pricing for demand response using online convex optimization," in *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*. IEEE, 2014, pp. 1–5.
- [12] V. Joseph and G. de Veciana, "Jointly optimizing multi-user rate adaptation for video transport over wireless systems: Mean-fairness-variability tradeoffs," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 567–575.
- [13] F. Zanini, D. Atienza, G. De Micheli, and S. P. Boyd, "Online convex optimization-based algorithm for thermal management of mpsoacs," in *Proceedings of the 20th symposium on Great lakes symposium on VLSI*. ACM, 2010, pp. 203–208.
- [14] M. Tanaka, "Real-time pricing with ramping costs: A new approach to managing a steep change in electricity demand," *Energy Policy*, vol. 34, no. 18, pp. 3634–3643, 2006.
- [15] R. Mookherjee, B. F. Hobbs, T. L. Friesz, and M. A. Rigdon, "Dynamic oligopolistic competition on an electric power network with ramping costs and joint sales constraints," *J. Ind. Manag. Optim.*, vol. 4, no. 3, pp. 425–452, 2008.
- [16] <https://www.misoenergy.org/MarketsOperations/RealTimeMarketData/Pages/DayAheadWindForecast.aspx>.
- [17] <http://www.pjm.com/planning/resource-adequacy-planning/load-forecast-dev-process.aspx>.
- [18] A. Rakhlin and K. Sridharan, "Online learning with predictable sequences," in *Conference on Learning Theory*, 2013, pp. 993–1019.
- [19] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online optimization: Competing with dynamic comparators," in *Artificial Intelligence and Statistics*, 2015, pp. 398–406.
- [20] D. Angeli, R. Amrit, and J. B. Rawlings, "On average performance and stability of economic model predictive control," *IEEE transactions on automatic control*, vol. 57, no. 7, pp. 1615–1626, 2012.

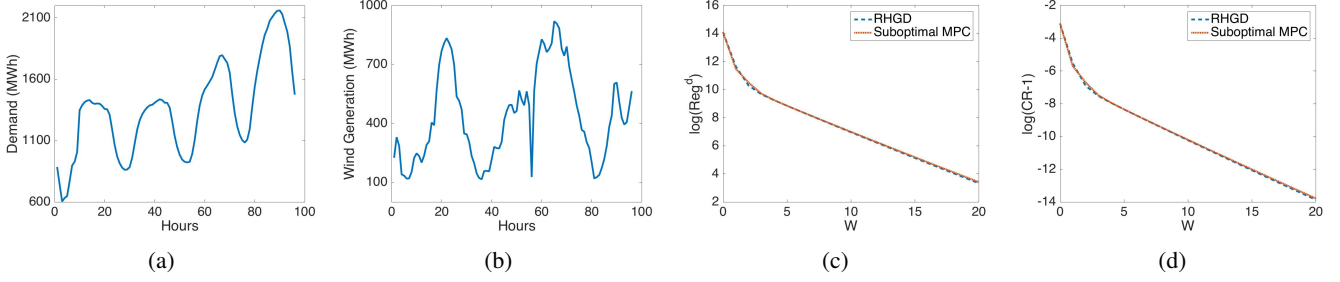


Fig. 1: (a) and (b) are demand and wind generation profile from New England ISO. (c) and (d) depicts the dynamic regret and competitive ratio of RHGD and suboptimal MPC in log scale.

- [21] L. Grüne and M. Stieler, “Asymptotic stability and transient optimality of economic mpc without terminal conditions,” *Journal of Process Control*, vol. 24, no. 8, pp. 1187–1196, 2014.
- [22] A. Ferramosca, D. Limon, and E. F. Camacho, “Economic mpc for a changing economic criterion for linear systems,” *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2657–2667, 2014.
- [23] M. Kögel and R. Findeisen, “Stabilization of inexact mpc schemes,” in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 5922–5928.
- [24] K. Graichen and A. Kugi, “Stability and incremental improvement of suboptimal mpc without terminal constraints,” *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2576–2580, 2010.
- [25] D. A. Allan, C. N. Bates, M. J. Risbeck, and J. B. Rawlings, “On the inherent robustness of optimal and suboptimal nonlinear mpc,” *Systems & Control Letters*, vol. 106, pp. 68–78, 2017.
- [26] J. Rawlings and D. Mayne, “Postface to model predictive control: Theory and design,” *Nob Hill Pub*, pp. 155–158, 2012.
- [27] M. Badieli, N. Li, and A. Wierman, “Online convex optimization with ramp constraints,” in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 6730–6736.
- [28] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman, “Using predictions in online optimization: Looking forward with an eye on the past,” in *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*. ACM, 2016, pp. 193–206.
- [29] R. Rosales and S. Sclaroff, “Improved tracking of multiple humans with trajectory prediction and occlusion modeling,” Boston University Computer Science Department, Tech. Rep., 1998.
- [30] O. Besbes, Y. Gur, and A. Zeevi, “Non-stationary stochastic optimization,” *Operations research*, vol. 63, no. 5, pp. 1227–1244, 2015.
- [31] <https://www.iso-ne.com/isoexpress/web/reports/operations/tree/daily-gen-fuel-type>.
- [32] <https://www.iso-ne.com/isoexpress/web/reports/load-and-demand/tree/zone-info>.
- [33] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

APPENDIX

A. Proof of Lemma 1

Proof. Remember that $C_1^T(x_1, \dots, x_T) = \sum_t f_t(x_t) + \frac{\beta}{2} \|x_t - x_{t-1}\|^2$. Since $\sum_t f_t(x_t)$ is α -strongly convex and smooth, we only need to study $\frac{\beta}{2} \|x_t - x_{t-1}\|^2$. The Hessian of switching cost is

$$A = \beta \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

and $0 \leq A \leq 4\beta I$. So $L = l + 4\beta$, and $C_1^T(x_1, \dots, x_T)$ is α -strongly convex. \square

B. Proof of Lemma 3

Before the formal proof, we introduce a supporting lemma, which upper bounds the switching cost of OGD outputs.

Lemma 4. Given $f_t \in \mathcal{F}_X(\alpha, l, G)$ for $t = 1, \dots, T$, and stepsize $\frac{1}{l}$, the outputs of OGD $\{x_t\}_{t=1}^T$ satisfy

$$\sum_{t=1}^T \|x_t - x_{t-1}\|^2 \leq \frac{2G}{l(1-\kappa)} \sum_{t=1}^T \|y_t^* - y_{t-1}^*\|$$

where x_1 is chosen to be $x_1 = x_0$, $\kappa = \sqrt{(1-\frac{\alpha}{l})}$, $y_t^* := \arg \min_{x_t \in X} f_t(x_t)$ for $1 \leq t \leq T$ and $y_0^* = x_0$.

Proof. Firstly, given $x_1 = x_0$, we have

$$\sum_{t=1}^T \|x_t - x_{t-1}\|^2 = \sum_{t=2}^T \|x_{t+1} - x_t\|^2$$

According to Corollary 2.3.2 in [33]

$$\sum_{t=1}^{T-1} \|x_{t+1} - x_t\|^2 \leq \frac{2}{l} \sum_{t=1}^{T-1} f_t(x_t) - f_t(y_t^*) \quad (11)$$

Since $f_t \in \mathcal{F}(\alpha, l, G)$, and $\|x_T - y_T^*\| \geq 0$,

$$\sum_{t=1}^{T-1} f_t(x_t) - f_t(y_t^*) \leq G \sum_{t=1}^{T-1} \|x_t - y_t^*\| \leq G \sum_{t=1}^T \|x_t - y_t^*\| \quad (12)$$

The remainder of the proof is to bound $\sum_{t=1}^T \|x_t - y_t^*\|$. By triangle inequality of Euclidean norm,

$$\sum_{t=1}^T \|x_t - y_t^*\| \leq \sum_{t=1}^T (\|x_t - y_{t-1}^*\| + \|y_t^* - y_{t-1}^*\|) \quad (13)$$

From Theorem 2.3.4 in [33], we have

$$\sum_{t=2}^T \|x_t - y_{t-1}^*\| \leq \kappa \sum_{t=1}^{T-1} \|x_t - y_t^*\|$$

where $\kappa = \sqrt{(1-\frac{\alpha}{\gamma})}$. Plug this in (13), we have

$$\begin{aligned} \sum_{t=1}^T \|x_t - y_t^*\| &\leq \kappa \sum_{t=1}^{T-1} \|x_t - y_t^*\| + \sum_{t=1}^T \|y_t^* - y_{t-1}^*\| \\ &\leq \kappa \sum_{t=1}^T \|x_t - y_t^*\| + \sum_{t=1}^T \|y_t^* - y_{t-1}^*\| \quad (14) \end{aligned}$$

where the first inequality is due to $x_1 = y_0^* = x_0$, and the second one is due to $\|x_T - y_T^*\| \geq 0$.

Regrouping the terms in (14) gives us:

$$\sum_{t=1}^T \|x_t - y_t^*\| \leq \frac{1}{1-\kappa} \sum_{t=1}^T \|y_t^* - y_{t-1}^*\| \quad (15)$$

This inequality together with (11) and (12) proves the bound in the lemma. \square

Now we are ready to prove Lemma 3.

Proof of Lemma 3:

$$\begin{aligned} \text{Reg}(OGD) &\leq \sum_{t=1}^T (f_t(x_t) - f_t(x_t^*) + \beta/2 \|x_t - x_{t-1}\|^2) \\ &\leq \sum_{t=1}^T (f_t(x_t) - f_t(y_t^*) + \beta/2 \|x_t - x_{t-1}\|^2) \\ &\leq G \sum_{t=1}^T \|x_t - y_t^*\| + \sum_{t=1}^T \|y_t^* - y_{t-1}^*\| \frac{G\beta}{l(1-\kappa)} \\ &\leq \frac{G}{(1-\kappa)} \sum_{t=1}^T \|y_t^* - y_{t-1}^*\| + \sum_{t=1}^T \|y_t^* - y_{t-1}^*\| \frac{G\beta}{l(1-\kappa)} \\ &= (\beta/l + 1) \frac{G}{(1-\kappa)} \sum_{t=1}^T \|y_t^* - y_{t-1}^*\| \end{aligned}$$

The first inequality is by throwing away negative term $-\beta/2 \|x_t^* - x_{t-1}^*\|^2$. The second one is because y_t^* minimizes $f_t(x_t)$. The third one is from bounded gradient and Lemma 4. The last one is by (15) and combining like terms. \square

C. Proof of Theorem 3

The proof of Theorem 3 relies on the following Lemma. It shows that, in the optimal offline solution, the dependence of the action x_t on the future y_{t+W}^* decays at most exponentially fast in W .

Lemma 5. *Given $y_t^* \in [-1, 1], t \in [T]$, the optimal offline solution can be written as $\mathbf{x}^* = \mathbf{A}\mathbf{y}^*$ where $\mathbf{y}^* = [y_1^*, y_2^*, \dots, y_T^*]'$, and \mathbf{A} is a T -by- T symmetric matrix. We write $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_T]'$, and $\mathbf{a}_t' = [a_{t,1}, a_{t,2}, \dots, a_{t,T}]$. Then, for $\tau \geq 0$, $a_{t,t+\tau}$ satisfies*

$$a_{t,t+\tau} \geq \frac{1}{\sqrt{4\frac{\beta}{\alpha} + 1}} \rho^\tau (1 - \rho^2 - 2\rho^{2T}). \quad (16)$$

Proof. Recall the offline optimization problem is given by,

$$\min_{x \in \mathbb{R}^T} \sum_{t=1}^T \left[\frac{\alpha}{2} (x_t - v_t)^2 + \frac{\beta}{2} (x_t - x_{t-1})^2 \right] \quad (17)$$

where x_1, \dots, x_T are the decision variables and $x_0 = 0$. Taking derivative of the cost function, we have (defining $\theta = \frac{\beta}{\alpha}$),

$$\begin{cases} x_t - v_t + \theta(x_t - x_{t-1}) + \theta(x_t - x_{t+1}) = 0, & t \in [T-1] \\ x_T - v_T + \theta(x_T - x_{T-1}) = 0 \end{cases} \quad (18)$$

Define

$$H = \begin{bmatrix} 1+2\theta & -\theta & 0 & 0 & \dots & 0 \\ -\theta & 1+2\theta & -\theta & 0 & \dots & 0 \\ 0 & -\theta & 1+2\theta & -\theta & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -\theta & 1+2\theta & -\theta \\ 0 & \dots & 0 & 0 & -\theta & 1+\theta \end{bmatrix}. \quad (19)$$

Then, (18) can be rewritten as $H\mathbf{x} = \mathbf{y}^*$. Since H is diagonally dominant, it is invertible. Also since, H 's diagonal entries are positive, while the off-diagonal entries are non-positive, we have H^{-1} is a nonnegative matrix. Let $\tilde{A} = H^{-1}$. Clearly \tilde{A} is symmetric since H is symmetric. Then we have,

$$\mathbf{x}^* = \mathbf{A}\mathbf{y}^*. \quad (20)$$

It remains to prove the lower bound on the entries of \tilde{A} . Define $\tilde{A} = [\tilde{a}_{i,j}]$ to be a T -by- T matrix, where $\tilde{a}_{t,t+\tau} = c\rho^{|\tau|}$ (for $t = 1, \dots, T, \tau = 1-t, \dots, T-t$). In what follows, we calculate $H\tilde{A}$. Firstly, for the second row till the second to last row, i.e. for $2 \leq t \leq T-1, 1-t \leq \tau \leq T-t$

$$\begin{aligned} [H\tilde{A}]_{t,t+\tau} &= -\theta\tilde{a}_{t-1,t+\tau} + (1+2\theta)\tilde{a}_{t,t+\tau} - \theta\tilde{a}_{t+1,t+\tau} \\ &= c[(1+2\theta)\rho^{|\tau|} - \theta\rho^{|\tau+1|} - \theta\rho^{|\tau-1|}] \\ &= \begin{cases} c\rho^\tau[(1+2\theta) - \theta\rho - \theta\frac{1}{\rho}] = 0 & \text{when } \tau \geq 1 \\ c\rho^{-\tau}[(1+2\theta) - \theta\rho - \theta\frac{1}{\rho}] = 0 & \text{when } \tau \leq -1 \\ c[(1+2\theta) - 2\theta\rho] = 1 & \text{when } \tau = 0 \end{cases} \end{aligned}$$

Then, we calculate the first row of $H\tilde{A}$. When $t = 1, 0 \leq \tau \leq T-1$,

$$\begin{aligned} [H\tilde{A}]_{1,1+\tau} &= (1+2\theta)\tilde{a}_{1,1+\tau} - \theta\tilde{a}_{2,1+\tau} \\ &= (1+2\theta)c\rho^{|\tau|} - \theta c\rho^{|\tau-1|} \\ &= \begin{cases} c\rho^\tau(1+2\theta - \theta\frac{1}{\rho}) & \text{when } \tau \geq 1 \\ c(1+2\theta - \theta\rho) & \text{when } \tau = 0 \end{cases} \\ &= \begin{cases} \rho^\tau[\frac{1}{2}(\frac{1}{\sqrt{4\theta+1}} - 1) + \frac{\theta}{\sqrt{4\theta+1}}] = \rho^\tau\rho\theta c & \text{when } \tau \geq 1 \\ \frac{1}{2}(\frac{1}{\sqrt{4\theta+1}} + 1) + \frac{\theta}{\sqrt{4\theta+1}} = 1 + \rho\theta c & \text{when } \tau = 0 \end{cases} \end{aligned}$$

For the last row, we have when $t = T, 1-T \leq \tau \leq 0$, we have

$$\begin{aligned} [H\tilde{A}]_{T,T+\tau} &= (1+\theta)\tilde{a}_{T,T+\tau} - \theta\tilde{a}_{T-1,T+\tau} \\ &= (1+\theta)c\rho^{|\tau|} - \theta c\rho^{|\tau+1|} \\ &= \begin{cases} c\rho^{-\tau}(1+\theta - \theta\frac{1}{\rho}) & \text{when } \tau \leq -1 \\ c(1+\theta - \theta\rho) & \text{when } \tau = 0 \end{cases} \\ &= \begin{cases} \rho^{-\tau}\frac{1}{2}(\frac{1}{\sqrt{4\theta+1}} - 1) = \rho^{-\tau}\frac{1}{2}(c-1) & \text{when } \tau \leq -1 \\ \frac{1}{2}(\frac{1}{\sqrt{4\theta+1}} + 1) = \frac{1}{2}(c+1) & \text{when } \tau = 0 \end{cases} \end{aligned}$$

Therefore, we have

$$I - H\tilde{A} = \begin{bmatrix} -\theta c\rho & -\theta c\rho^2 & -\theta c\rho^3 & \dots & -\theta c\rho^T \\ 0 & 0 & 0 & \cdot & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdot & 0 \\ \frac{1}{2}(1-c)\rho^{T-1} & \frac{1}{2}(1-c)\rho^{T-2} & \frac{1}{2}(1-c)\rho^{T-3} & \dots & \frac{1}{2}(1-c) \end{bmatrix}$$

Recall the t 'th row of A is \mathbf{a}'_t . We also let the t 'th row of \tilde{A} be $\tilde{\mathbf{a}}'_t$. Let \mathbf{e}_t be the T -dimensional vector with the t 'th entry being 1 and other entries being 0. Then, $\mathbf{a}_t = A\mathbf{e}_t$ and $\tilde{\mathbf{a}}_t = \tilde{A}\mathbf{e}_t$. Then, we have,

$$\begin{aligned}\mathbf{a}_t - \tilde{\mathbf{a}}_t &= A\mathbf{e}_t - \tilde{A}\mathbf{e}_t = A(I - H\tilde{A})\mathbf{e}_t \\ &= A[-\theta c\rho^t, 0, \dots, 0, \frac{1}{2}(1-c)\rho^{T-t}]' \\ &= -\theta c\rho^t \mathbf{a}_1 + \frac{1}{2}(1-c)\rho^{T-t} \mathbf{a}_T\end{aligned}\quad (21)$$

Now we set $t = 1$ and $t = T$ in (21), and get

$$\mathbf{a}_1 - \tilde{\mathbf{a}}_1 = -\theta c\rho \mathbf{a}_1 + \frac{1}{2}(1-c)\rho^{T-1} \mathbf{a}_T \quad (22)$$

$$\mathbf{a}_T - \tilde{\mathbf{a}}_T = -\theta c\rho^T \mathbf{a}_1 + \frac{1}{2}(1-c)\mathbf{a}_T \quad (23)$$

Manipulating (23), and we can get

$$\mathbf{a}_T = \frac{2}{1+c} \tilde{\mathbf{a}}_T - \rho^T \frac{2\theta c}{1+c} \mathbf{a}_1 \leq \frac{2}{1+c} \tilde{\mathbf{a}}_T \quad (24)$$

where we have used the fact that all the entries of \mathbf{a}_1 are nonnegative. Manipulating (22), using $1 + \rho\theta c = \frac{\theta c}{\rho}$ and plugging (24) into (22), we have,

$$\begin{aligned}\mathbf{a}_1 &= \frac{1}{1+c\theta\rho} \tilde{\mathbf{a}}_1 + \frac{1-c}{2(1+c\theta\rho)} \rho^{T-1} \mathbf{a}_T \\ &= \frac{\rho}{\theta c} \tilde{\mathbf{a}}_1 + (1-c) \frac{1}{2\theta c} \rho^T \mathbf{a}_T \\ &\leq \frac{\rho}{\theta c} \tilde{\mathbf{a}}_1 + \frac{1-c}{1+c} \frac{1}{\theta c} \rho^T \tilde{\mathbf{a}}_T\end{aligned}\quad (25)$$

Plugging the above into (24),

$$\begin{aligned}\mathbf{a}_T &= \frac{2}{1+c} \tilde{\mathbf{a}}_T - \rho^T \frac{2\theta c}{1+c} \mathbf{a}_1 \\ &\geq \frac{2}{1+c} \tilde{\mathbf{a}}_T - \rho^T \frac{2\theta c}{1+c} \left(\frac{\rho}{\theta c} \tilde{\mathbf{a}}_1 + \frac{1-c}{1+c} \frac{1}{\theta c} \rho^T \tilde{\mathbf{a}}_T \right) \\ &= \frac{2}{1+c} \tilde{\mathbf{a}}_T - \frac{2}{1+c} \rho^{T+1} \tilde{\mathbf{a}}_1 - \frac{1-c}{1+c} \frac{2}{1+c} \rho^{2T} \tilde{\mathbf{a}}_T \\ &\geq \frac{2}{1+c} \tilde{\mathbf{a}}_T - \frac{2}{1+c} \rho^{T+1} \tilde{\mathbf{a}}_1 - \frac{2}{1+c} \rho^{2T} \tilde{\mathbf{a}}_T\end{aligned}\quad (26)$$

Using the upper bound on \mathbf{a}_1 (25) and the lower bound on \mathbf{a}_T (26) and plugging them into (21), we have

$$\begin{aligned}\mathbf{a}_t - \tilde{\mathbf{a}}_t &= -\theta c\rho^t \mathbf{a}_1 + \frac{1}{2}(1-c)\rho^{T-t} \mathbf{a}_T \\ &\geq -\theta c\rho^t \left[\frac{\rho}{\theta c} \tilde{\mathbf{a}}_1 + \frac{1-c}{1+c} \frac{1}{\theta c} \rho^T \tilde{\mathbf{a}}_T \right] \\ &\quad + \frac{1}{2}(1-c)\rho^{T-t} \left[\frac{2}{1+c} \tilde{\mathbf{a}}_T - \frac{2}{1+c} \rho^{T+1} \tilde{\mathbf{a}}_1 - \frac{2}{1+c} \rho^{2T} \tilde{\mathbf{a}}_T \right] \\ &= -\rho^{t+1} \tilde{\mathbf{a}}_1 - \frac{1-c}{1+c} \rho^{t+T} \tilde{\mathbf{a}}_T \\ &\quad + \frac{1-c}{1+c} \rho^{T-t} \tilde{\mathbf{a}}_T - \frac{1-c}{1+c} \rho^{2T-t+1} \tilde{\mathbf{a}}_1 - \frac{1-c}{1+c} \rho^{3T-t} \tilde{\mathbf{a}}_T \\ &\geq -\rho^{t+1} \tilde{\mathbf{a}}_1 - \rho^{2T-t+1} \tilde{\mathbf{a}}_1 - \rho^{3T-t} \tilde{\mathbf{a}}_T\end{aligned}$$

Therefore, for $1 \leq t \leq T$ and $0 \leq \tau \leq T-t$,

$$\begin{aligned}a_{t,t+\tau} &\geq c\rho^\tau - \rho^{t+1} c\rho^{t+\tau-1} - \rho^{2T-t+1} c\rho^{t+\tau-1} - \rho^{3T-t} c\rho^{T-t-\tau} \\ &= c\rho^\tau - c\rho^{2t+\tau} - c\rho^{2T+\tau} - c\rho^{4T-2t-2\tau}\end{aligned}$$

$$\begin{aligned}&= c\rho^\tau (1 - \rho^{2t} - \rho^{2T} - \rho^{4T-2t-2\tau}) \\ &\geq c\rho^\tau (1 - \rho^2 - 2\rho^{2T})\end{aligned}$$

where in the last inequality we have used $\rho^{2t} \leq \rho^2$, and by $t + \tau \leq T$, $\rho^{4T-2t-2\tau} \leq \rho^{2T}$. \square

We are now ready to prove Theorem 3.

Proof of Theorem 3: For $t \leq T - W$, define $\mathbf{b}_t = [a_{t,1}, a_{t,2}, \dots, a_{t,t+W-1}, 0, \dots]'$; for $t \geq T - W + 1$, define $\mathbf{b}_t = \mathbf{a}_t$. Define $\mathbf{c}_t = \mathbf{a}_t - \mathbf{b}_t$ (\mathbf{c}'_t is the t 'th row of A with the first $t + W - 1$ entries being set to 0), and $C = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_T]'$.

First we fix $\mathcal{A}_1, \dots, \mathcal{A}_T$. We generate y_t^* i.i.d. from a Bernoulli distribution, with $P(y_t^* = 1) = P(y_t^* = -1) = 1/2$. We then have for $t \leq T - W$,

$$\begin{aligned}(x_t - x_t^*)^2 &= (x_t - \mathbf{b}'_t \mathbf{y}^* - \mathbf{c}'_t \mathbf{y}^*)^2 \\ &= (x_t - \mathbf{b}'_t \mathbf{y}^*)^2 + (\mathbf{c}'_t \mathbf{y}^*)^2 - 2(\mathbf{c}'_t \mathbf{y}^*)(x_t - \mathbf{b}'_t \mathbf{y}^*) \\ &\geq (\mathbf{c}'_t \mathbf{y}^*)^2 + 2(\mathbf{c}'_t \mathbf{y}^*)(\mathbf{b}'_t \mathbf{y}^* - x_t)\end{aligned}$$

We take expectation conditioned on $y_1^*, y_2^*, \dots, y_{t+W-1}^*$

$$\begin{aligned}\mathbb{E}[(x_t - x_t^*)^2 | y_1^*, \dots, y_{t+W-1}^*] &\stackrel{(a)}{\geq} \mathbb{E}[(\mathbf{c}'_t \mathbf{y}^*)^2 | y_1^*, \dots, y_{t+W-1}^*] \\ &\quad + 2\mathbb{E}[(\mathbf{c}'_t \mathbf{y}^*) | y_1^*, \dots, y_{t+W-1}^*](\mathbf{b}'_t \mathbf{y}^* - x_t) \\ &\stackrel{(b)}{\geq} \mathbb{E}[(\mathbf{c}'_t \mathbf{y}^*)^2 | y_1^*, \dots, y_{t+W-1}^*] + 2\mathbb{E}[(\mathbf{c}'_t \mathbf{y}^*)](\mathbf{b}'_t \mathbf{y}^* - x_t) \\ &\stackrel{(c)}{=} \mathbb{E}[(\mathbf{c}'_t \mathbf{y}^*)^2 | y_1^*, \dots, y_{t+W-1}^*]\end{aligned}$$

where (a) is because $\mathbf{b}'_t \mathbf{y}^*$ and x_t only depends on $y_1^*, \dots, y_{t+W-1}^*$; (b) is because $\mathbf{c}'_t \mathbf{y}^*$ only depends on y_{t+W}^*, \dots, y_T^* , which are independent from $y_1^*, \dots, y_{t+W-1}^*$; (c) is because y_τ^* has expectation 0 for any τ . Now we take expectation again over $y_1^*, \dots, y_{t+W-1}^*$, and get $\mathbb{E}[(x_t - x_t^*)^2] \geq \mathbb{E}[(\mathbf{c}'_t \mathbf{y}^*)^2]$. Summing over t ,

$$\mathbb{E}\|\mathbf{x} - \mathbf{x}^*\|^2 \geq \sum_{t=1}^{T-W} \mathbb{E}(\mathbf{c}'_t \mathbf{y}^*)^2 = \mathbb{E}\|C\mathbf{y}^*\|^2 = \mathbb{E}(\mathbf{y}^*)' C' C \mathbf{y}^*$$

Now since $C'C$ is a T -by- T symmetric positive semi-definite matrix, we diagonalize $C'C$ to get $C'C = \sum_{t=1}^T \lambda_t \zeta_t \zeta'_t$, where λ_t is an eigenvalue of $C'C$ associated with eigenvector ζ_t , and ζ_1, \dots, ζ_T form an orthonormal basis of \mathbb{R}^T . Thus,

$$\mathbb{E}((\mathbf{y}^*)' C' C \mathbf{y}^*) = \sum_{t=1}^T \mathbb{E}(\lambda_t (\mathbf{y}^*)' \zeta_t \zeta'_t \mathbf{y}^*) = \sum_{t=1}^T \lambda_t \mathbb{E}(\zeta'_t \mathbf{y}^*)^2$$

Let $\zeta_t = [\zeta_{t,1}, \dots, \zeta_{t,T}]'$, then $\mathbb{E}(\zeta'_t \mathbf{y}^*)^2 = \text{Var}(\zeta'_t \mathbf{y}^*) = \sum_{\tau=1}^t \zeta_{t,\tau}^2 \text{Var}(y_\tau^*) = 1$. Therefore, $\mathbb{E}\|\mathbf{x} - \mathbf{x}^*\|^2 \geq \sum_{t=1}^T \lambda_t = \text{Trace}(C'C) = \sum_{t=1}^T \|\mathbf{c}_t\|^2$. By the definition of \mathbf{c}_t , and (16), for $t \leq T - W$,

$$\|\mathbf{c}_t\|^2 \geq a_{t,t+W}^2 \geq \frac{\alpha(1-\rho^2)^2}{4(4\beta + \alpha)} \rho^{2W}$$

where we have used by the lower bound on T , $2\rho^{2T} \leq \frac{1-\rho^2}{2}$. As a result,

$$\mathbb{E}\|\mathbf{x} - \mathbf{x}^*\|^2 \geq (T - W) \frac{\alpha(1-\rho^2)^2}{4(4\beta + \alpha)} \rho^{2W}.$$

Therefore, there must exist a realization of \mathbf{y}^* , s.t.

$$\begin{aligned}\|\mathbf{x} - \mathbf{x}^*\|^2 &\geq (T - W) \frac{\alpha(1 - \rho^2)^2}{4(4\beta + \alpha)} \rho^{2W} \\ &\geq \frac{T - W}{T} \frac{\alpha(1 - \rho^2)^2}{8(4\beta + \alpha)} \rho^{2W} \sum_{t=1}^T \|y_t^* - y_{t-1}^*\|\end{aligned}$$

where in the last inequality, we have used $\sum_{t=1}^T \|y_t^* - y_{t-1}^*\| \leq 2T$. The results of the theorem follows directly from the α -strong convexity of the off-line cost function and $T \geq 2W$. \square