On the Value of Look-Ahead in Competitive Online Convex Optimization

MING SHI, Purdue University, USA
XIAOJUN LIN, Purdue University, USA
LEI JIAO, University of Oregon, USA

Although using look-ahead information is known to improve the competitive ratios of online convex optimization (OCO) problems with switching costs, the competitive ratios obtained from existing results often depend on the cost coefficients of the problem, and can potentially be large. In this paper, we propose new online algorithms that can utilize look-ahead to achieve much lower competitive ratios for OCO problems with switching costs and hard constraints. For the perfect look-ahead case where the algorithm is provided with the exact inputs in a future look-ahead window of size K, we design an Averaging Regularized Moving Horizon Control (ARMHC) algorithm that can achieve a competitive ratio of $\frac{K+1}{K}$. To the best of our knowledge, ARMHC is the first to attain a low competitive ratio that is independent of either the coefficients of the switching costs and service costs, or the upper and lower bounds of the inputs. Then, for the case when the future look-ahead has errors, we develop a Weighting Regularized Moving Horizon Control (WRMHC) algorithm that carefully weights the decisions inside the look-ahead window based on the accuracy of the look-ahead information. As a result, WRMHC also achieves a low competitive ratio that is independent of the cost coefficients, even with uncertain hard constraints. Finally, our analysis extends online primal-dual analysis to the case with look-ahead by introducing a novel "re-stitching" idea, which is of independent interest.

CCS Concepts: • **Theory of computation** → **Online algorithms**; • **Mathematics of computing** → *Convex optimization*; • **Networks** → Network algorithms.

Additional Key Words and Phrases: Online convex optimization (OCO); Look-ahead; Competitive analysis; Online primal-dual analysis

ACM Reference Format:
Ming Shi, Xiaojun Lin, and Lei Jiao. 2019. On the Value of Look-Ahead in Competitive Online Convex Optimization. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 2, Article 22 (June 2019), 42 pages. https://doi.org/10.1145/3326136

1 INTRODUCTION

In this paper, we study online convex optimization (OCO) problems with switching costs and hard constraints [3, 4, 19, 20, 28, 34]. In its generic form, an OCO problem proceeds in time $t = 1, 2, ..., \mathcal{T}$. At each time t, first the environment (or adversary) reveals the input $\vec{A}(t) \in \mathbb{R}^{M\times 1}$ and the service-cost function $h_t(\cdot, \vec{A}(t))$. Then, the decision maker chooses the decision $\vec{X}(t) \in \mathbb{R}^{N\times 1}$, which is based

Authors' addresses: Ming Shi, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47906, USA, sming@purdue.edu; Xiaojun Lin, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47906, USA, linx@ecn.purdue.edu; Lei Jiao, Department of Computer and Information Science, University of Oregon, Eugene, OR, 97403, USA, jiao@cs.uoregon.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Association for Computing Machinery.
2476-1249/2019/6-ART22 $15.00
https://doi.org/10.1145/3326136

on all revealed inputs from $\vec{A}(1)$ to $\vec{A}(t)$. A service cost $h_t(\vec{X}(t), \vec{A}(t))$ is incurred for the decision $\vec{X}(t)$. Further, there is a switching cost that penalizes the increments $[\vec{X}(t) - \vec{X}(t-1)]^+$ for each time t, where $[x]^+ = \max\{x, 0\}$. In the majority of this paper, we focus on the case where the service cost is linear in $\vec{X}(t)$ and the switching cost is linear in the increment $[\vec{X}(t) - \vec{X}(t-1)]^+$, although some of the results will also be generalized to convex service costs. Such an OCO formulation can be used to model online decisions in many application domains, such as machine learning [12, 27, 30, 34], networking [20, 21, 23, 28], cloud computing [7, 8, 22, 29], cyber-physical systems [1, 11, 19, 24] and wireless systems [18, 25, 31]. (See specific examples of applications in Sec. 2.1.) Moreover, in many practical settings, the decision $\vec{X}(t)$ is subject to hard constraints that must be satisfied at each time t. For example, enough VNFs need to be set up to serve the incoming traffic [9, 28]. In an offline setting, the entire input sequence $\vec{A}(1)$ to $\vec{A}(\mathcal{T})$ was known in advance. Then, the optimal offline decisions could be found by solving a simple convex optimization problem. However, we are interested in the online setting where $\vec{A}(t)$ is only revealed at time t (along with some limited look-ahead information to be introduced shortly). Our goal is then to design online algorithms with low competitive ratios, where the competitive ratio is defined as the worst-case ratio between the total costs of an online algorithm and that of the optimal offline solution.

Although there exist a number of online algorithms that can achieve provable competitive ratios for fairly general classes of OCO problems [3, 7, 8, 20, 28], it remains a challenge to achieve low competitive ratios. For example, the regularization method can be used on very general classes of OCO problems without any future information [3, 14]. It computes the online decision at each time by replacing the immediate switching-cost with a carefully-chosen strictly-convex term that more aggressively penalizes large increments of the control variables. Although a provable competitive ratio can be shown, the resulting competitive ratio often depends on the dimension of the problem and the characteristics of the inputs. For example, the reference [14] shows that, for an NFV orchestration and scaling problem, the competitive ratio is $\log\left(1 + \frac{MI}{\epsilon}\right) + 1 + \frac{1}{\phi}$, where M is the number of VNFs, I is the number of servers, and ϕ is the lower bound of any positive decision variables. As a result, the competitive ratio can potentially be large. This is not surprising because, when there is absolutely no future knowledge, the competitive ratio often has to be large due to adversarial inputs (unless further restrictions are imposed on the specific problem structure [1, 11]). Fortunately, in many applications, such as resource allocation in data centers [20, 28], power dispatching in smart grid [11, 32, 33], and video streaming in wireless systems [18, 25, 31], the future input can be predicted to some extent. Such future information can be modeled as a look-ahead window [20]. Intuitively, the further ahead such prediction is provided, and the more accurate the prediction is, the lower the competitive ratio should be. It is thus of great interest in understanding how to best utilize either perfect or imperfect look-ahead to achieve low competitive ratios.

Online decisions using perfect look-ahead have been studied in [6, 20, 26]. However, existing results are still unsatisfactory due to potentially large competitive ratios. Receding Horizon Control (RHC), also known as Model Predictive Control (MPC), is a popular framework in the literature to use look-ahead [6, 20, 26]. RHC optimizes the total costs in the look-ahead window based on the predicted inputs, and then only commits to the first decision. However, even in the perfect look-ahead case, it has been shown that the competitive ratio of RHC may not improve as the look-ahead window size K increases [8, 20]. Recently, Lin et al. propose the Averaging Fixed Horizon Control (AFHC), which averages decisions from multiple rounds of Fixed Horizon Control (FHC). AFHC is shown to be $(1 + \max_n \frac{w_n}{(K+1)\cdot c_n})$-competitive, where c_n is the service-cost coefficient and w_n is the switching-cost coefficient [20]. Note that while this competitive ratio of AFHC improves

as K increases, it still increases with $\max_n \frac{w_n}{c_n}$. In contrast, note that the competitive ratios of the regularization method [3] does not depend on $\frac{w_n}{c_n}$. Therefore, it remains an open question whether one can develop online algorithms that can achieve low competitive ratios both decreasing in K and independent of w_n and c_n for the OCO problem that we consider in this paper.

For imperfect look-ahead, existing results mainly focus on the competitive difference of RHC, AFHC, and CHC (Committed Horizon Control) [7, 8]. Note that the competitive difference is the *difference* between the cost of an online algorithm and that of the optimal offline solution, while the competitive ratio is the *ratio* between them. Thus, even if the competitive difference is bounded, it does not imply that the competitive ratio is bounded. In fact, even the competitive difference in [7, 8] still depends on such problem parameters as the coefficients of the switching costs. In addition, there are also crucial differences between the problem formulations of [7, 8] and that in this paper. First, the model in [7] and [8] assumes a probability distribution of the uncertainty (i.e., an auto-regressive model driven by *i.i.d.* signals). In practice, it may be difficult to know what exactly the distribution of uncertainty is. In contrast, we are interested in the worst-case competitive guarantees, without assuming a distribution of uncertainty. Second, these existing results typically do not consider hard constraints that depend on the input (e.g., the service must be greater than the input demand). The difficulty in combining such hard constraints with imperfect look-ahead is that such hard constraints become *uncertain* for future time slots in the look-ahead window. As a result, decisions based on imperfect look-ahead may not satisfy the actual hard constraints in the future. The work in [11] studies competitive online algorithms under such uncertain hard constraints, but for a more specific problem structure where the switching cost is incurred for only one decision variable. Recently, we propose a Robust Affine Policy (RAP) that optimizes the competitive ratio for OCO problems within the class of affine policies [28]. Although RAP can work with hard constraints, its competitive ratio still depends on the switching-cost coefficient w_n [28]. Thus, for the general OCO problems that we study in this paper, it remains an open problem to develop online algorithms with low competitive ratios under imperfect look-ahead and uncertain hard constraints.

In summary, none of the existing results can achieve low competitive ratios independent of the switching-cost and service-cost coefficients under either perfect look-ahead or imperfect look-ahead. In this paper, we design new online algorithms that address these open questions. We make the following contributions.

(1) First, we study the case with perfect look-ahead, i.e., at time t, inputs from $\vec{A}(t)$ to $\vec{A}(t+K)$ are given to the decision maker. We first develop a novel Averaging Regularized Moving Horizon Control (ARMHC) algorithm that is $(\frac{K+1}{K})$-competitive. We note that this result is highly appealing because the competitive ratio depends only on the size of the look-ahead window K, but not on other problem parameters, including both the coefficients of the service costs and switching costs, and the upper and lower bounds of future inputs. To the best of our knowledge, this is the first result of such type in the literature for general OCO problems with perfect look-ahead. Further, note that even with a small look-ahead window of size $K = 1$, the competitive ratio is already 2. In contrast, the competitive ratio of the regularization method is often much larger [3]. This suggests that even a small window of perfect look-ahead can be highly effective in reducing the competitive ratio. Moreover, as the look-ahead window size K increases, the competitive ratio of ARMHC further decreases to 1.

In order to develop this result, we introduce in ARMHC two key ideas. One is to use a "regularization term" at the tail of each look-ahead window, and the other is to commit to only the first $K-1$ decisions in the look-ahead window. As we will elaborate in Sec. 3.2, both ideas are crucial for achieving competitive ratios much lower than the state-of-art. While our proof utilizes ideas

from online primal-dual analysis [5], a key difficulty is to ensure that, even with look-ahead, the online dual variables are still feasible for the offline problem (see Sec. 3.3). To address this difficulty, we introduce a novel "re-stitching" idea to deal with the gap between look-ahead rounds, which is of independent interest.

(2) Second, we study the case with imperfect look-ahead, i.e., at time t, only a possible trajectory of the near-future inputs from $\vec{A}(t+1)$ to $\vec{A}(t+K)$ is given to the decision maker, as well as upper and lower bounds of the possible future inputs. Our proposed variant of RMHC will use the upper bounds of the future inputs in the decisions so that future hard constraints are met after averaging multiple decisions. However, note that the accuracy of future predictions typically varies in time, e.g., the prediction accuracy is usually lower further into the future. Thus, even within a look-ahead window, the qualities of decisions made by RMHC are different for different time slots, i.e., the decision for a time slot will be less reliable if the prediction accuracy for the time slot is lower. Due to this reason, simply averaging multiple rounds of RMHC is no longer a good idea. Instead, we propose to assign different weights to the costs and decisions at different time slots and across different rounds of RMHC. By carefully choosing the weights, our proposed Weighting Regularized Moving Horizon Control (WRMHC) algorithm can achieve an even lower competitive ratio under imperfect look-ahead. Finally, the competitive ratio of WRMHC is also independent of the service-cost and switching-cost coefficient.

We note that the reference [13] also develops randomized online algorithms for a fairly general class of "benefit task systems", which can attain a competitive ratio of $\frac{K+1}{K}$ under perfect look-ahead window of size K. The idea behind the randomized online algorithms proposed in [13] also shares some similarity to the averaging idea in AFHC and ARMHC. However, there is a crucial difference. [13] considers the problem of *maximizing* a non-negative benefit. In contrast, we (as well as the related work in [3, 7, 8, 14, 20]) consider the problem of *minimizing* a non-negative cost. For benefit-maximization problems, the loss of benefit due to an online decision at a given time slot is upper-bounded by the corresponding benefit gained by the optimal offline decision. In contrast, for cost-minimization problems, the switching cost incurred by an online decision at a given time slot can potentially be much larger than the corresponding cost of the optimal offline decision. Due to this reason, online versions of the benefit-maximization problems are more tractable than online versions of the cost-minimization problems, and thus the competitive results of [13] do not apply to our cost-minimization setting either. To the best of our knowledge, our paper is the first one that can attain a low $\frac{K+1}{K}$ competitive ratio under perfect look-ahead for cost-minimization problems.

2 PROBLEM FORMULATION

In this section, we provide the formulation of the online convex optimization (OCO) problem that we considered in this paper.

2.1 OCO problem with switching costs and hard constraints

Consider a time-horizon of $\mathcal{T}$ time slots. At each time $t = 1, 2, ..., \mathcal{T}$, first the input $\vec{A}(t) = [a_m, m = 1, ..., M]^{\mathrm{T}} \in \mathbb{R}^{M\times 1}$ and a service-cost function $h_t(\cdot, \vec{A}(t))$ are revealed ($[\cdot]^{\mathrm{T}}$ denotes the transpose of a vector or matrix). Then, the decision maker chooses a decision $\vec{X}(t) = [x_n(t), n = 1, ..., N]^{\mathrm{T}} \in \mathbb{R}^{N\times 1}$ from a convex decision set $\mathbb{X}_t(\vec{A}(t))$. The set $\mathbb{X}_t(\vec{A}(t))$ may contain hard constraints that depend on input, and can be written as linear inequalities in $(\vec{X}(t), \vec{A}(t))$, i.e.,

$$\boldsymbol{B_1}\vec{X}(t) \geq \boldsymbol{B_2}\vec{A}(t), \text{ for all time } t = 1, ..., \mathcal{T}, \tag{1}$$

where $\boldsymbol{B}_1$ is an $L \times N$ matrix and $\boldsymbol{B}_2$ is an $L \times M$ matrix. We assume that $\boldsymbol{B}_1, \boldsymbol{B}_2 \geq 0$, where "$\geq$" is component-wise. For example, one such constraint could be $\sum_{n=1}^{N} x_n(t) \geq \sum_{m=1}^{M} a_m(t)$, which can mean that the incoming traffic must be served by the servers in data centers [20, 28] or the power demand must be satisfied by the dispatching generation in smart grids [32].

Provided that the hard constraint is met at time t, a service cost $h_t(\vec{X}(t), \vec{A}(t))$ is incurred. For the moment, we assume that the service cost is linear, i.e., $h_t(\vec{X}(t), \vec{A}(t)) = \vec{C}^{\mathrm{T}}(t)\vec{X}(t)$, where $\vec{C}(t) = [c_n(t), n = 1, ..., N]^{\mathrm{T}} \in \mathbb{R}^{N\times 1}$ is the service-cost coefficient. (Our results for the perfect look-ahead case will also be generalized to convex service costs in Sec. 3.4.) Note that if the hard constraint is not satisfied, we assume that the service cost will be $+\infty$. Additionally, there is a switching cost $\vec{W}^{\mathrm{T}}[\vec{X}(t) - \vec{X}(t-1)]^+$ that penalizes the increment[1] of each entry of the decision $\vec{X}(t)$ at time t, where $\vec{W} = [w_n, n = 1, ..., N]^{\mathrm{T}} \in \mathbb{R}^{N\times 1}$ is the switching-cost coefficient.

Remark 1. Although the above linearity assumptions may seem restrictive, they can be used to model many practical applications. For example, in the Network Function Virtualization (NFV) orchestration and scaling problems [9, 14, 28], the service cost corresponds to the cost of setting up functions at servers and the distance cost of routing traffic, which are both linear in the control variables (i.e., the number of functions set up at each server and the fraction of traffic routed to each server). The switching cost can model the cost of migrating the states from one server to another server, which is proportional to the change of control variables. Similarly, in cloud or edge computing [15–17], the service cost corresponds to the operation cost of the servers or the energy cost, which are linear in the control variables (i.e., the number of servers that are used). The switching cost captures the costs for hardware wear-and-tear, state migration and profile loading. These costs are also proportional to the changes of decision variables. Further, note that we have assumed the switching-cost coefficients to be time-invariant. In practice, this is often a reasonable assumption, e.g., the switching cost in NFV corresponds to the cost of migrating the states from one server to another server. This cost is proportional to the amount of states that need to be migrated, and the cost-coefficient does not change in time [14, 28]. A similar assumption on time-invariant switching-cost coefficients has also been made in [1, 7, 8, 11, 20].

If all the future inputs were known, then the above problem would have been a standard convex optimization problem as follows,

$$\min_{\{\vec{X}(1),\ldots,\vec{X}(\mathcal{T})\}} \left\{ \sum_{t=1}^{\mathcal{T}} \vec{C}^{\mathrm{T}}(t)\vec{X}(t) + \sum_{t=1}^{\mathcal{T}} \vec{W}^{\mathrm{T}}[\vec{X}(t) - \vec{X}(t-1)]^+ \right\} \tag{2a}$$

$$\text{sub. to: } \boldsymbol{B}_1\vec{X}(t) \geq \boldsymbol{B}_2\vec{A}(t), \text{ for all time } t \in [1, \mathcal{T}], \tag{2b}$$

$$\vec{X}(t) \geq 0, \text{ for all time } t \in [1, \mathcal{T}], \tag{2c}$$

where $[l_1, l_2]$ denotes the set $\{l_1, l_1 + 1, ..., l_2\}$ and $\vec{X}(0) = 0$ as typically assumed in the literature of OCO problems [8]. However, in this paper (as in [8, 12, 27, 30, 34]), we focus on the setting where the decision maker must make decisions in an *online* fashion. This means that at each time t, she must make the current decision $\vec{X}(t)$ based on only the current input $\vec{A}(t)$ and a limited amount of future look-ahead (which will be defined precisely in Sec. 2.2). Moreover, once the decision $\vec{X}(t)$ is made, the decision is irrevocable. In future time slots $t + 1, t + 2, ..., \mathcal{T}$, she cannot go back and revise the decision $\vec{X}(t)$. Thus, this problem becomes an *online* problem.

[1]Note that some related papers on OCO problems charge the switching cost by the absolute value, e.g., $|\vec{X}(t) - \vec{X}(t-1)|$. It has been shown in [4] that switching costs defined by absolute values can be converted to switching costs defined by increments, because the sum of the absolute values of the changes is upper-bounded by twice the sum of the increments.

For ease of exposition, We use $\vec{C}(t_1 : t_2)$ to collect the service-cost coefficients $\vec{C}(t)$ from $t = t_1$ to t_2, i.e., $\vec{C}(t_1 : t_2) \triangleq \left\{\vec{C}(t), \text{ for all } t \in [t_1, t_2]\right\}$. Define $\vec{A}(t_1 : t_2)$ and $\vec{X}(t_1 : t_2)$ similarly. We now introduce the look-ahead model.

2.2 Modeling Look-Ahead

As we discussed in the introduction, in many applications the future inputs can be predicted to some extent. We can model such partial future information via a look-ahead window. Let the size of the look-ahead window be $K \geq 1$. At each time t, the decision maker will not only know the exact input $\vec{A}(t)$ of time t, but also know some information about the future inputs $\vec{A}(t+1)$, $\vec{A}(t+2)$, ..., $\vec{A}(t+K)$. Note that the future inputs after time $t+K$ are still unknown to the decision maker at time t. Next, we differentiate between perfect and imperfect look-ahead.

(i) In the perfect look-ahead case, at each time t, the decision maker can know the *precise* values of future service-cost coefficients $\vec{C}(t+1 : t+K)$ and inputs $\vec{A}(t+1 : t+K)$ within the look-ahead window. Thus, the online algorithm can make the current decision $\vec{X}(t)$ based on not only all revealed service-cost coefficients $\vec{C}(1 : t)$, inputs $\vec{A}(1 : t)$ and past decisions $\vec{X}(1 : t-1)$, but also future values of $\vec{C}(t+1 : t+K)$ and $\vec{A}(t+1 : t+K)$ in the look-ahead window.

(ii) In the imperfect look-ahead case, at each time t, we assume that the decision maker still knows the *precise* future service-cost coefficients $\vec{C}(t+1 : t+K)$ within the look-ahead window. However, she only knows approximate values of future inputs $\vec{A}(t+1 : t+K)$ in the look-ahead window [11]. Specifically, she is given a predicted trajectory $\vec{A}_t^{\text{pred}}(t+1 : t+K) = \left\{a_{t,m}^{\text{pred}}(t+i), m \in [1, M], i \in [1, K]\right\}$ from time $t+1$ to $t+K$, as well as bounds on how far the real inputs $\vec{A}(t+1 : t+K)$ can deviate from the predicted trajectory. Specifically, at a future time $t+i$, $i = 1, 2, ..., K$, the ratio of the m-th component of the real input to that of the predicted input is bounded by

$$f_m^{\text{low}}(i) \leq \frac{a_m(t+i)}{a_{t,m}^{\text{pred}}(t+i)} \leq f_m^{\text{up}}(i), \text{for all } m \in [1, M], i \in [1, K]. \tag{3}$$

The upper-bounds $\vec{f}^{\text{up}}(i) = \{f_m^{\text{up}}(i), \text{for all } m \in [1, M]\}$ and the lower-bounds $\vec{f}^{\text{low}}(i) = \{f_m^{\text{low}}(i), \text{for all } m \in [1, M]\}$ are functions of i and are known in advance. Notice that for current time t, i.e., $i = 0$, $\vec{f}^{\text{up}}(0) = \vec{f}^{\text{low}}(0) = 1$, since the current input $\vec{A}(t)$ has already been revealed. Thus, the online algorithm needs to make the current decision $\vec{X}(t)$ based on not only all revealed inputs $\vec{A}(1 : t)$ and past decisions $\vec{X}(1 : t-1)$, but also the look-ahead information, given by $\vec{A}_t^{\text{pred}}(t+1 : t+K)$, $\vec{f}^{\text{up}}(1 : K) = \{\vec{f}^{\text{up}}(i), \text{ for all } i \in [1, K]\}$ and $\vec{f}^{\text{low}}(1 : K) = \{\vec{f}^{\text{low}}(i), \text{ for all } i \in [1, K]\}$. Notice that if $f_m^{\text{up}}(i) \equiv 1$ and $f_m^{\text{low}}(i) \equiv 1$ for all $m \in [1, M]$ and $i \in [1, K]$, then $\vec{A}_t^{\text{pred}}(t+i)$ is exactly the real input $\vec{A}(t+i)$ for all $i \in [1, K]$ in the look-ahead window, and we thus reduce to the case of perfect look-ahead.

Remark 2. The assumption that the service-cost coefficients $\vec{C}(t+1 : t+K)$ are still known in the imperfect look-ahead case is reasonable in certain applications. For example, the service-cost coefficients may be fixed in advance for the entire time-horizon, as is the case for the distance costs and the setup costs in NFV [14, 28], or the operation cost and energy cost (when the energy price is fixed) in cloud or edge computing [15–17].

As we discussed briefly in Sec. 1, for imperfect look-ahead, the hard constraints (1) are more challenging to deal with. This is because the hard constraints in (1) for future time slots are based on the predicted input $\vec{A}_t^{\text{pred}}(t+1 : t+K)$ and thus become uncertain. As a result, decisions based

on imperfect prediction may not be able to meet the real constraints in the future. Hence, these *uncertain* hard constraints in the look-ahead window need to be considered carefully.

2.3 The Performance Metric

For an online algorithm π, let $\text{Cost}^{\pi}(1:\mathcal{T})$ be its total cost, i.e.,

$$\text{Cost}^{\pi}(1:\mathcal{T}) = \sum_{t=1}^{\mathcal{T}} \left\{ \vec{C}^{\text{T}}(t)\vec{X}^{\pi}(t) + \vec{W}^{\text{T}}[\vec{X}^{\pi}(t) - \vec{X}^{\pi}(t-1)]^{+} \right\}, \tag{4}$$

where $\vec{X}^{\pi}(t)$ is the decision of the online algorithm π at each time t. Let $\vec{X}^{\text{OPT}}(1:\mathcal{T})$ be the optimal offline solution to the optimization problem (2), whose total cost is $\text{Cost}^{\text{OPT}}(1:\mathcal{T})$. We evaluate an online algorithm π using its competitive ratio

$$\text{CR}^{\pi} \triangleq \max_{\{\text{all possible inputs } \vec{A}(1:\mathcal{T})\}} \frac{\text{Cost}^{\pi}(1:\mathcal{T})}{\text{Cost}^{\text{OPT}}(1:\mathcal{T})}, \tag{5}$$

i.e., the worst-case ratio of its total cost to that of the optimal offline solution. Thus, in this paper we are interested in developing online algorithms with low competitive ratios.

3 THE PERFECT LOOK-AHEAD CASE

In this section, we consider first the case with perfect look-ahead. In [20], Averaging Fixed Horizon Control (AFHC) has been proposed as a way to utilize such perfect look-ahead information. It was shown in [20] that AFHC can achieve lower competitive ratios than Receding Horizon Control (RHC) for certain classes of online problems. However, in the following we will show via a simple counter-example that AFHC could still incur large competitive ratios. This counter-example thus motivates us to develop new competitive online algorithms that can achieve significantly lower competitive ratios with perfect look-ahead.

3.1 A Counter-Example for AFHC

Algorithm 1 Averaging Fixed Horizon Control (AFHC)

Input: $\vec{A}(1:\mathcal{T})$, K.
Output: $\vec{X}^{\text{AFHC}}(1:\mathcal{T})$.
FOR $t = -K+1:\mathcal{T}$
Step 1: Let $\tau \leftarrow t \bmod (K+1)$, and $\vec{X}(t-1) \leftarrow \vec{X}^{\text{FHC}^{(\tau)}}(t-1)$.
Step 2: Based on $\vec{A}(t:t+K)$, solve problem $Q^{\text{FHC}^{(\tau)}}(t)$ in Eq.(7) to get $\vec{X}^{\text{FHC}^{(\tau)}}(t:t+K)$.
Step 3: Use Eq.(6) below to compute the final decision

$$\vec{X}^{\text{AFHC}}(t) = \frac{1}{K+1} \sum_{\tau=0}^{K} \vec{X}^{\text{FHC}^{(\tau)}}(t). \tag{6}$$

END

We first describe the behavior of AFHC from [20], which is given in Algorithm 1 for completeness. AFHC with a look-ahead window of size K takes the average of the decisions of $K+1$ versions of Fixed Horizon Control (FHC) algorithms as follows. Let τ be an integer from 0 to K. The τ-th version of FHC, denoted by $\text{FHC}^{(\tau)}$, calculates decisions at time $t = \tau + (K+1)\cdot u$, where $u = -1, 0, \ldots, \left\lfloor \frac{\mathcal{T}}{K+1} \right\rfloor$. Specifically, at each time $t = \tau + (K+1)\cdot u$, the current input $\vec{A}(t)$ and inputs

in the current look-ahead window $t+1, \ldots, t+K$ are revealed. $\text{FHC}^{(\tau)}$ then calculates the solution to the following problem $Q^{\text{FHC}^{(\tau)}}(t)$ at each time $t = \tau + (K+1) \cdot u$,

$$\begin{aligned} &\min_{\vec{X}(t:t+K)} \left\{ \sum_{s=t}^{t+K} \left[\vec{C}^{\mathrm{T}}(s)\vec{X}(s) + \vec{W}^{\mathrm{T}}[\vec{X}(s) - \vec{X}(s-1)]^{+} \right] \right\} \\ &\text{sub. to: } \boldsymbol{B_1}\vec{X}(s) \geq \boldsymbol{B_2}\vec{A}(s), \text{ for all time } s \in [t, t+K], \\ &\vec{X}(s) \geq 0, \text{ for all time } s \in [t, t+K]. \end{aligned} \tag{7}$$

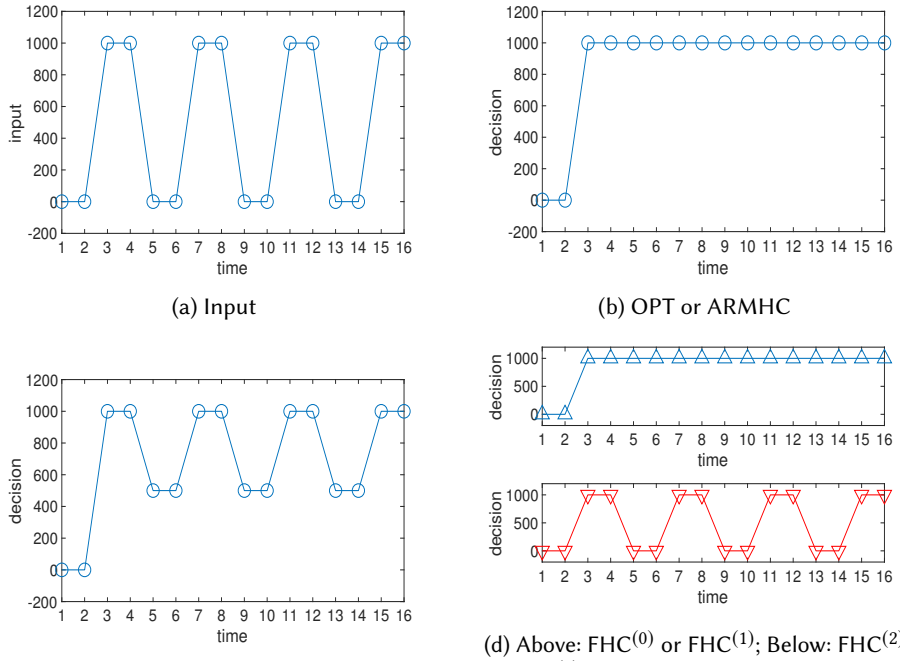
Here, the initial value $\vec{X}(t-1)$ is also from $\text{FHC}^{(\tau)}$, but is computed at an earlier time $t-K-1$, i.e., by solving $Q^{\text{FHC}^{(\tau)}}(t-K-1)$ as in Eq.(7). Further, whenever the time index s is outside the set $[1, \mathcal{T}]$, we use the convention that $\vec{A}(s) = 0$ and $\vec{X}(s) = 0$. By concatenating multiple rounds of $\text{FHC}^{(\tau)}$ at time $t = \tau + (K+1) \cdot u$, $u = -1, 0, \ldots, \left\lfloor \frac{\mathcal{T}}{K+1} \right\rfloor$, we get an entire decision sequence denoted by $\vec{X}^{\text{FHC}^{(\tau)}}(1:\mathcal{T})$. Then, AFHC simply takes the average of $\vec{X}^{\text{FHC}^{(\tau)}}(1:\mathcal{T})$ for all $\tau \in [0, K]$.

We briefly explain the intuition why AFHC can achieve a competitive ratio decreasing in the look-ahead window size K. Recall that the decision of AFHC at time t is the average of K versions of FHC from time $t-K$ to t. For versions of FHC from time $t-K+1$ to t, they already know the future input at time $t+1$ thanks to perfect look-ahead. Hence, it is intuitive that their switching costs from time t to $t+1$ should be close to optimal. However, for the version of FHC at time $t-K$, since its look-ahead window ends at time t, it does not know the future input at time $t+1$. As a result, its "end-of-look-ahead" decision may potentially lead to larger-than-optimal switching cost from time t to $t+1$. Fortunately, because AFHC takes the average of all of them, the undesirable effect due to the version of FHC at time $t-K$ is reduced by a factor of K. This is the reason that the competitive ratio of AFHC decreases with K. (In contrast, the competitive ratio of RHC or FHC may not decrease with K [20].) However, the magnitude of this undesirable effect still depends on the coefficients of the switching costs. Therefore, the competitive ratio of AFHC will also depend on the coefficients of the switching costs, and can still be quite large, especially when K is not very large. In the following example, we demonstrate that AFHC can indeed incur an arbitrarily large competitive ratio.

Counter-example 1: We use a look-ahead window of size $K = 3$ as an example, but similar behaviors can arise for any value of K. Assume that both $\vec{X}(t)$ and $\vec{A}(t)$ are 1-dimension, and thus can be replaced by scalars $x(t)$ and $a(t)$. Further, the hard constraint is $x(t) \geq a(t)$ for all time t. There are two possible values, $\overline{a}, \underline{a} \geq 0$, for the inputs. The high value $\overline{a}$ is assumed to be much larger than the low value $\underline{a}$. In this example, the input sequence is $a(4j-3) = a(4j-2) = \underline{a}$ and $a(4j-1) = a(4j) = \overline{a}$, $j = 1, 2, \ldots, \frac{\mathcal{T}}{4}$, which is plotted in Fig. 1a for $\underline{a} = 0$ and $\overline{a} = 1000$. The switching-cost coefficient w is much larger than the service-cost coefficient $c(t)$. For ease of explanation, we assume that $\mathcal{T}$ is a multiple of 4 here.

Cost of the optimal offline solution: Since the offline algorithm knows the entire input sequence in advance, it sees that the high input $\overline{a}$ will appear in the future. Thus, the optimal offline solution will set the decision $x^{\text{OPT}}(t) = \underline{a}$ for time $t = 1, 2$ and $x^{\text{OPT}}(t) = \overline{a}$ for all time t later (even when $a(t) = \underline{a}$) to avoid switching costs altogether. The decisions are plotted in Fig. 1b for $\underline{a} = 0$ and $\overline{a} = 1000$. The corresponding total cost is $\text{Cost}^{\text{OPT}}(1:\mathcal{T}) = [c(1) + c(2)]\,\underline{a} + w(\overline{a} - \underline{a}) + \sum_{t=3}^{\mathcal{T}} c(t)\overline{a}$. (The second term is the only switching cost, which is incurred at time $t = 3$.)

Cost of AFHC: Since AFHC does not have all the future information, it is difficult for AFHC to emulate the optimal offline decision. To see this, note that for $K = 3$, there are 4 versions of FHC. The decisions of $\text{FHC}^{(0)}$, $\text{FHC}^{(1)}$, $\text{FHC}^{(2)}$ and $\text{FHC}^{(3)}$ are plotted in Fig. 1d for $\underline{a} = 0$



(a) Input

(b) OPT or ARMHC

(c) AFHC

(d) Above: $\text{FHC}^{(0)}$ or $\text{FHC}^{(1)}$; Below: $\text{FHC}^{(2)}$ or $\text{FHC}^{(3)}$.

Fig. 1. Behaviors of OPT, AFHC and ARMHC in Counter-example 1.

and $\overline{a} = 1000$. Although $\text{FHC}^{(0)}$ and $\text{FHC}^{(1)}$ follow exactly the same decisions as the optimal offline solution, $\text{FHC}^{(2)}$ and $\text{FHC}^{(3)}$ will have a different behavior due to the "end-of-look-ahead" problem, i.e., they will choose a lower value of $\vec{X}(t)$ in the later part of the look-ahead window because the last input of the look-ahead window is $\underline{a}$. Please see Appendix A for the derivation of the decision sequences of different versions of $\text{FHC}^{(\tau)}$. Taking the average of these decisions, AFHC gives the final decision sequence $(\underline{a}, \underline{a}, \overline{a}, \overline{a})$ for the first round and $(\frac{\overline{a}+\underline{a}}{2}, \frac{\overline{a}+\underline{a}}{2}, \overline{a}, \overline{a})$ for all subsequent rounds starting from time 5. The decision sequence is plotted in Fig. 1c for $\underline{a} = 0$ and $\overline{a} = 1000$. (See Appendix A for details.) The corresponding total cost is $\text{Cost}^{\text{AFHC}}(1:\mathcal{T}) = [c(1)+c(2)]\,\underline{a} + [c(3)+c(4)]\,\overline{a} + w(\overline{a}-\underline{a}) + \sum_{j=2}^{\mathcal{T}/4}\left\{\left[c(4j-3)+c(4j-2)\right]\frac{\overline{a}+\underline{a}}{2} + \left[c(4j-1)+c(4j)\right]\overline{a} + w\frac{\overline{a}-\underline{a}}{2}\right\}$.

In summary, we can observe that the total switching cost of AFHC increases by $w\frac{\overline{a}-\underline{a}}{2}$ every 4 time slots due to the terms $\sum_{j=2}^{\mathcal{T}/4} w\frac{\overline{a}-\underline{a}}{2}$, while the switching cost incurred by the optimal offline solution is only $w\overline{a}$ for the entire time horizon. If $w \geq \sum_{t=1}^{\mathcal{T}} c(t)$, the total-cost ratio of AFHC and the optimal offline solution is

$$\frac{\text{Cost}^{\text{AFHC}}(1:\mathcal{T})}{\text{Cost}^{\text{OPT}}(1:\mathcal{T})} \geq \frac{\frac{\mathcal{T}}{4} w \frac{\overline{a}-\underline{a}}{2}}{2w\overline{a}} = \frac{\mathcal{T}}{16}\frac{\overline{a}-\underline{a}}{\overline{a}}, \tag{8}$$

which can become arbitrarily large as the time horizon $\mathcal{T}$ increases.

3.2 Averaging Regularized Moving Horizon Control

From Counter-example 1, we can see that the competitive ratio of AFHC depends on the service-cost coefficients and the switching-cost coefficients, and could be large. In this section, we will introduce an Averaging Regularized Moving Horizon Control (ARMHC) algorithm, which can achieve a competitive ratio independent of the switching-cost and service-cost coefficients.

Analogous to AFHC, ARMHC takes the average of the decisions from multiple rounds of a subroutine, called Regularized Moving Horizon Control (RMHC). Fig. 2 provides an overview of RMHC and ARMHC. Specifically, let the size of the look-ahead window be K, and τ be an integer from 0 to $K-1$. The τ-th version of RMHC, denoted by $\text{RMHC}^{(\tau)}$, plays a similar role as $\text{FHC}^{(\tau)}$, but with some crucial differences. At each time $t = \tau + Ku$, $u = -1, 0, ..., \lfloor \frac{\mathcal{T}}{K} \rfloor$, recall that the current input $\vec{A}(t)$ is revealed. Further $\text{RMHC}^{(\tau)}$ is given the inputs in the look-ahead window $t+1, ..., t+K$. $\text{RMHC}^{(\tau)}$ then calculates the solution to the following problem $Q^{\text{RMHC}^{(\tau)}}(t)$ at each time $t = \tau + Ku$,

$$\min_{\vec{X}(t:t+K-1)} \left\{ \sum_{s=t}^{t+K-1} \left[\vec{C}^{\text{T}}(s)\vec{X}(s) + \vec{W}^{\text{T}}[\vec{X}(s) - \vec{X}(s-1)]^{+} \right] - [\vec{W}^{\text{T}} - \vec{C}^{\text{T}}(t+K)]^{+} \cdot \vec{X}(t+K-1) \right\} \tag{9a}$$

$$\text{sub. to: } \boldsymbol{B}_1\vec{X}(s) \geq \boldsymbol{B}_2\vec{A}(s), \text{ for all time } s \in [t, t+K-1], \tag{9b}$$

$$\vec{X}(s) \geq 0, \text{ for all time } s \in [t, t+K-1]. \tag{9c}$$

Note that unlike $\text{FHC}^{(\tau)}$, which produces $K+1$ slots of decisions for the current time and the entire look-ahead window, $\text{RMHC}^{(\tau)}$ only produces the first K slots of decisions. Due to this reason, the initial value $\vec{X}(t-1)$ for $Q^{\text{RMHC}^{(\tau)}}(t)$ is from $\text{RMHC}^{(\tau)}$ computed at an earlier time of $t-K$ (rather than $t-K-1$ for $\text{FHC}^{(\tau)}$), where $Q^{\text{RMHC}^{(\tau)}}(t-K)$ was solved. In other words, the multiple rounds of $\text{RMHC}^{(\tau)}$ repeat every K time slots. As in FHC, we also follow the convention that $\vec{A}(t) = 0$ and $\vec{X}(t) = 0$ for all $t \leq 0$ and $t > \mathcal{T}$. In this way, multiple rounds of $\text{RMHC}^{(\tau)}$ concatenated also produce an entire decision sequence $\vec{X}^{\text{RMHC}^{(\tau)}}(1:\mathcal{T})$. Then, ARMHC takes the average of $\vec{X}^{\text{RMHC}^{(\tau)}}(1:\mathcal{T})$ for all $\tau \in [0, K-1]$. The details are given in Algorithm 2. Note that if $t+K > \mathcal{T}$, the tail of RMHC exceeds the time horizon $\mathcal{T}$. These last rounds of RMHC thus have completed future information, and can solve an offline problem $\tilde{Q}^{\text{RMHC}^{(\tau)}}(t)$ in Eq.(10) instead of $Q^{\text{RMHC}^{(\tau)}}(t)$ in Eq.(9).

Now we discuss the key properties of ARMHC. Comparing ARMHC with AFHC, the main difference is in the formulation of problem $Q^{\text{RMHC}^{(\tau)}}(t)$ in Eq.(9) versus the formulation of problem $Q^{\text{FHC}^{(\tau)}}(t)$ in Eq.(7). We can see two important ideas in RMHC.

(i) *Go-Back-1-Step:* In Eq.(9), instead of optimizing the decisions over all $K+1$ time slots of the look-ahead window (as in Eq.(7)), RMHC only optimizes the first K time slots of decisions. The intuition is that the last time-slot's decision is too difficult to choose since the future is unknown.

(ii) *Regularization at the tail:* In Algorithm 2, at each time t, there is an additional term, i.e., $-[\vec{W}^{\text{T}} - \vec{C}^{\text{T}}(t+K)]^{+} \cdot \vec{X}(t+K-1)$, at the end of the objective in Eq.(9). This term can be viewed as a "regularization term" on the last decision $\vec{X}(t+K-1)$. Such a regularization term aims to prevent unnecessary changes between time $t+K-1$ and time $t+K$. For example, suppose that there exists an index $n \in [1, N]$, such that $c_n(t+K) < w_n$. Since the service cost is small, the optimal offline decision $x_n(t+K)$ may be large at time $t+K$. Then, in order to avoid large switching costs, the optimal offline decision may also use a larger value of $x_n(t+K-1)$ at time $t+K-1$. This effect is emulated by RMHC with the above regularization term. To see this, note that if $c_n(t+K) < w_n$, we

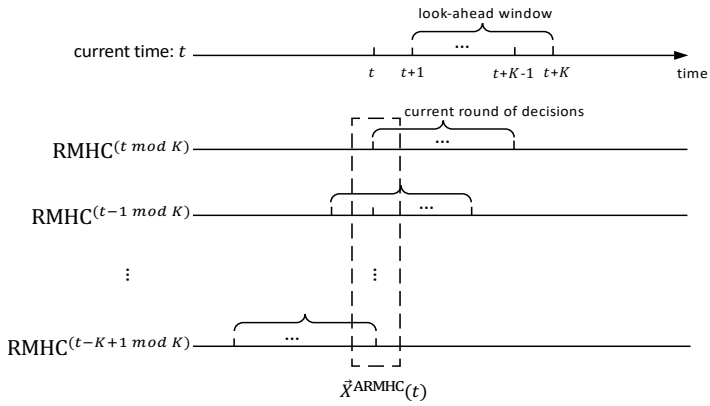


Fig. 2. Averaging Regularized Moving Horizon Control: at each time t, it averages decisions from K versions of RMHC$^{(\tau)}$.

Algorithm 2 Averaging Regularized Moving Horizon Control

Input: $\vec{A}(1:\mathcal{T})$, K.
Output: $\vec{X}^{\text{ARMHC}}(1:\mathcal{T})$.
FOR $t = -K+2:\mathcal{T}$
Step 1: $\tau \leftarrow t \bmod K$, and then $\vec{X}(t-1) \leftarrow \vec{X}^{\text{RMHC}^{(\tau)}}(t-1)$.
Step 2: Given the current input $\vec{A}(t)$ and the near-term future inputs $\vec{A}(t+1:t+K)$ in the look-ahead window,
if $t+K \leq \mathcal{T}$ **then**
solve $Q^{\text{RMHC}^{(\tau)}}(t)$ in Eq.(9) to get $\vec{X}^{\text{RMHC}^{\tau}}(t:t+K-1)$.
else
solve $\tilde{Q}^{\text{RMHC}^{(\tau)}}(t)$ in Eq.(10) to get $\vec{X}^{\text{RMHC}^{(\tau)}}(t:\mathcal{T})$.

$$\min_{\vec{X}(t:\mathcal{T})} \left\{ \sum_{s=t}^{\mathcal{T}} \left[\vec{C}^{\text{T}}(s)\vec{X}(s) + \vec{W}^{\text{T}}[\vec{X}(s) - \vec{X}(s-1)]^{+} \right] \right\} \tag{10a}$$

$$\text{sub. to: } \boldsymbol{B_1}\vec{X}(s) \geq \boldsymbol{B_2}\vec{A}(s), \text{ for all time } s \in [t, \mathcal{T}], \tag{10b}$$

$$\vec{X}(s) \geq 0, \text{ for all time } s \in [t, \mathcal{T}]. \tag{10c}$$

end if
Step 3: Use Eq.(11) below to get the final decision

$$\vec{X}^{\text{ARMHC}}(t) = \frac{1}{K} \sum_{\tau=0}^{K-1} \vec{X}^{\text{RMHC}^{(\tau)}}(t). \tag{11}$$

END

have $-[w_n - c_n(t+K)]^{+} < 0$. Then, to minimize the total objective function of $Q^{\text{RMHC}^{(\tau)}}(t)$ in Eq.(9), RMHC may also set $x_n(t+K-1)$ to be a larger value. On the other hand, when $c_n(t+K) \geq w_n$, the optimal offline decision $x_n(t+K)$ is likely to be small, and thus we no longer need $x_n(t+K-1)$ to be large. This effect is also emulated by RMHC with the above regularization term. To see this, note that in this case $[w_n - c_n(t+K)]^{+} = 0$. Then, the regularization term plays no roles in determining

the value of $x_n(t+K-1)$. We will defer to Sec. 3.3 (see Remark 4) to explain how we construct this regularization term in this particular way.

Remark 3. We note that a similar idea of "Go-Back-1-Step" has appeared in the literature. For example, in [8], the authors propose to compute the decisions for the entire look-ahead window, but only commit to the first part of the decisions. However, in Appendix B, we will provide a counter-example to show that just adding the "Go-Back-1-Step" idea to AFHC will still lead to poor competitive ratios. Thus, the second idea of "regularization" is also critical. It will be clear from our analysis in Sec. 3.3 that it relies on the combination of these two ideas to works. Finally, we note that our regularization term is different from most results in the literature [3].

Cost of ARMHC: For the input sequence in Counter-example 1 in Sec. 3.1, each version of RMHC will always set the decision $x(t) = \overline{a}$ after time 2. This is because the regularization term contains the large switching-cost coefficient. Thus, even if the last input of the look-ahead window is $\underline{a}$, RMHC will choose a large value of $\vec{X}(t+K-1)$ to minimize the objective in Eq.(9). The decisions at time $t=1$ and time $t=2$ are obviously equal to $\underline{a}$. By taking the average of 3 versions of RMHC$^{(\tau)}$, ARMHC gives the final decision $x(1) = x(2) = \underline{a}$ and $x(t) = \overline{a}$ for all time $t > 2$. The final decision sequence of ARMHC will be the same as the optimal offline solution that is plotted in Fig. 1b.

3.3 Competitive Analysis

The following Theorem 3.1 provides the theoretical performance guarantee for the competitive ratio of ARMHC.

Theorem 3.1. *Given a look-ahead window of size K, the Averaging Regularized Moving Horizon Control (ARMHC) algorithm is $(\frac{K+1}{K})$-competitive, i.e., for any $\mathcal{T}$,*

$$Cost^{ARMHC}(1:\mathcal{T}) \leq \frac{K+1}{K} Cost^{OPT}(1:\mathcal{T}), \text{ for all } A(1:\mathcal{T}). \tag{12}$$

As shown in Theorem 3.1, the competitive ratio of ARMHC is only a function of the look-ahead window size K, and is independent of any cost coefficients or bounds on future inputs. In contrast, as we demonstrated earlier, the competitive ratio of AFHC [20] could increase arbitrarily as the switching-cost coefficients increase. Similarly, the competitive ratio of the regularization method [3] could be arbitrarily large if the gap between the upper and lower bounds of future inputs increases. To the best of our knowledge, ARMHC is the first algorithm in the literature that can utilize perfect look-ahead to attain such a clean and parameter-independent competitive ratio.

The rest of the section is devoted to the proof of Theorem 3.1. We first give the high-level idea, starting from a typical online primal-dual analysis. For the offline problem (2), by introducing an auxiliary variable $\vec{Y}(t)$ for the switching term $[\vec{X}(t) - \vec{X}(t-1)]^+$, we can get an equivalent formulation of the offline optimization problem (2),

$$\min_{\{\vec{X}(1:\mathcal{T}), \vec{Y}(1:\mathcal{T})\}} \left\{ \sum_{t=1}^{\mathcal{T}} \vec{C}^{\mathrm{T}}(t)\vec{X}(t) + \sum_{t=1}^{\mathcal{T}} \vec{W}^{\mathrm{T}}\vec{Y}(t) \right\} \tag{13a}$$

$$\text{sub. to: } \boldsymbol{B_1}\vec{X}(t) \geq \boldsymbol{B_2}\vec{A}(t), \text{ for all time } t \in [1, \mathcal{T}], \tag{13b}$$

$$\vec{Y}(t) \geq \vec{X}(t) - \vec{X}(t-1), \text{ for all time } t \in [1, \mathcal{T}], \tag{13c}$$

$$\vec{X}(t), \vec{Y}(t) \geq 0, \text{ for all time } t \in [1, \mathcal{T}]. \tag{13d}$$

Then, let $\vec{\beta}(t) = [\beta_l(t), l = 1, ..., L]^{\mathrm{T}} \in \mathbb{R}^{L\times 1}$ and $\vec{\theta}(t) = [\theta_n(t), n = 1, ..., N]^{\mathrm{T}} \in \mathbb{R}^{N\times 1}$ be the Lagrange multipliers for constraints (13b) and (13c), respectively. We have the offline dual optimization

problem to (13),

$$\max_{\{\vec{\beta}(1:\mathcal{T}),\vec{\theta}(1:\mathcal{T})\}} \sum_{t=1}^{\mathcal{T}} \vec{\beta}^{\mathrm{T}}(t)\boldsymbol{B}_2\vec{A}(t) \tag{14a}$$

$$\text{sub. to: } \vec{\beta}(t) \geq 0, \vec{\theta}(t) \geq 0, \text{ for all time } t \in [1,\mathcal{T}], \tag{14b}$$

$$\vec{C}(t) - \boldsymbol{B}_1^{\mathrm{T}}\vec{\beta}(t) + \vec{\theta}(t) - \vec{\theta}(t+1) \geq 0, \text{ for all time } t \in [1,\mathcal{T}], \tag{14c}$$

$$\vec{W} - \vec{\theta}(t) \geq 0, \text{ for all time } t \in [1,\mathcal{T}]. \tag{14d}$$

Let $D^{\mathrm{OPT}}(1:\mathcal{T})$ and $\mathrm{Cost}^{\mathrm{OPT}}(1:\mathcal{T})$ be the optimal offline dual objective value for (14) and the optimal offline primal cost for (13), respectively. According to the duality theorem [2, p. 260], we know that

$$D^{\mathrm{OPT}}(1:\mathcal{T}) = \mathrm{Cost}^{\mathrm{OPT}}(1:\mathcal{T}). \tag{15}$$

The key of proving Theorem 3.1 is to show that RMHC produces a sequence of online dual variables that are feasible to the offline dual optimization problem (14). Then, since the dual problem is a maximization problem, the corresponding online dual cost, denoted by $D^{\mathrm{RMHC}}(1:\mathcal{T})$, satisfies $D^{\mathrm{RMHC}}(1:\mathcal{T}) \leq D^{\mathrm{OPT}}(1:\mathcal{T})$. If we can show that the primal and dual costs of RMHC satisfy $\mathrm{Cost}^{\mathrm{RMHC}}(1:\mathcal{T}) \leq \frac{K+1}{K} \cdot D^{\mathrm{RMHC}}(1:\mathcal{T})$, the desired result then follows.

However, along this path, we have two difficulties. First, because of the additional regularization terms in the objective of RMHC, the relationship between $\mathrm{Cost}^{\mathrm{RMHC}}(1:\mathcal{T})$ and $D^{\mathrm{RMHC}}(1:\mathcal{T})$ needs to be revisited. Second, the dual variables computed by different rounds of RMHC may not be feasible for the offline problem. Below, we will address these difficulties step-by-step.

Step-1 (Quantifying the gap between the primal cost and the dual cost of RMHC): First, similar to the transformation in (13), we can get an equivalent formulation of problem (9),

$$\min_{\{\vec{X}(t:t+K-1),\vec{Y}(t:t+K-1)\}} \left\{ \sum_{s=t}^{t+K-1} \left[\vec{C}^{\mathrm{T}}(s)\vec{X}(s) + \vec{W}^{\mathrm{T}}\vec{Y}(s)\right] - [\vec{W}^{\mathrm{T}} - \vec{C}^{\mathrm{T}}(t+K)]^{+} \cdot \vec{X}(t+K-1) \right\} \tag{16a}$$

$$\text{sub. to: } \boldsymbol{B}_1\vec{X}(s) \geq \boldsymbol{B}_2\vec{A}(s), \text{ for all time } s \in [t, t+K-1], \tag{16b}$$

$$\vec{Y}(s) \geq \vec{X}(s) - \vec{X}(s-1), \text{ for all time } s \in [t, t+K-1], \tag{16c}$$

$$\vec{X}(s), \vec{Y}(s) \geq 0, \text{ for all time } s \in [t, t+K-1]. \tag{16d}$$

For each round of $\mathrm{RMHC}^{(\tau)}$ from time t to $t+K-1$, we define the primal cost

$$\begin{aligned}\mathrm{Cost}^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1) = \sum_{s=t}^{t+K-1} \Big\{ &\vec{C}^{\mathrm{T}}(s)\vec{X}^{\mathrm{RMHC}^{(\tau)}}(s) \\ &+ \vec{W}^{\mathrm{T}}[\vec{X}^{\mathrm{RMHC}^{(\tau)}}(s) - \vec{X}^{\mathrm{RMHC}^{(\tau)}}(s-1)]^{+} \Big\}.\end{aligned} \tag{17}$$

Note that $\mathrm{Cost}^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1)$ includes the switching cost from time $t-1$ to t but not the switching cost from time $t+K-1$ to $t+K$.

Next, let $\left(\vec{\beta}^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1), \vec{\theta}^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1)\right)$ be the optimal dual solution to the dual of the optimization problem (16). Then, for each $\tau \in [0, K-1]$, we can define the online dual cost in a similar manner as of the offline dual problem (14), i.e.,

$$D^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1) \triangleq \sum_{s=t}^{t+K-1} \left(\vec{\beta}^{\mathrm{RMHC}^{(\tau)}}(s)\right)^{\mathrm{T}} \boldsymbol{B}_2\vec{A}(s). \tag{18}$$

$$G^{\text{old}} \triangleq \begin{bmatrix} D^{\text{RMHC}^{(0)}}(-K:-1) & D^{\text{RMHC}^{(0)}}(0:K-1) & \cdots & D^{\text{RMHC}^{(0)}}(\lfloor \frac{\mathcal{T}}{K} \rfloor \cdot K : \lfloor \frac{\mathcal{T}}{K} \rfloor \cdot K + K - 1) \\ D^{\text{RMHC}^{(1)}}(-K+1:0) & D^{\text{RMHC}^{(1)}}(1:K) & \cdots & D^{\text{RMHC}^{(1)}}(\lfloor \frac{\mathcal{T}}{K} \rfloor \cdot K + 1 : \lfloor \frac{\mathcal{T}}{K} \rfloor \cdot K + K) \\ \vdots & \vdots & \vdots & \vdots \\ D^{\text{RMHC}^{(K-1)}}(-1:K-2) & D^{\text{RMHC}^{(K-1)}}(K-1:2K-2) & \cdots & D^{\text{RMHC}^{(K-1)}}(\lfloor \frac{\mathcal{T}}{K} \rfloor \cdot K + K - 1 : \lfloor \frac{\mathcal{T}}{K} \rfloor \cdot K + 2K - 2) \end{bmatrix} \tag{21}$$

$$G^{\text{new}} \triangleq \begin{bmatrix} D^{\text{RMHC}^{(0)}}(-K:-1) & 0 & D^{\text{RMHC}^{(1)}}(1:K) & 0 & \cdots & D^{\text{RMHC}^{(\tau_1(\mathcal{U}))}}(1+\mathcal{U}\cdot(K+1):K+\mathcal{U}\cdot(K+1)) \\ D^{\text{RMHC}^{(1)}}(-K+1:0) & 0 & D^{\text{RMHC}^{(2)}}(2:K+1) & 0 & \cdots & D^{\text{RMHC}^{(\tau_2(\mathcal{U}))}}(2+\mathcal{U}\cdot(K+1):K+1+\mathcal{U}\cdot(K+1)) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ D^{\text{RMHC}^{(K-1)}}(-1:K-2) & 0 & D^{\text{RMHC}^{(0)}}(K:2K-1) & 0 & \cdots & D^{\text{RMHC}^{(\tau_K(\mathcal{U}))}}(K+\mathcal{U}\cdot(K+1):2K-1+\mathcal{U}\cdot(K+1)) \\ D^{\text{RMHC}^{(0)}}(0:K-1) & 0 & D^{\text{RMHC}^{(1)}}(K+1:2K) & 0 & \cdots & D^{\text{RMHC}^{(\tau_{K+1}(\mathcal{U}))}}(K+1+\mathcal{U}\cdot(K+1):2K+\mathcal{U}\cdot(K+1)) \end{bmatrix} \tag{22}$$

However, note that (16) contains additional regularization terms in the objective function. Thus, there will be some gap between $\text{Cost}^{\text{RMHC}^{(\tau)}}(t:t+K-1)$ and $D^{\text{RMHC}^{(\tau)}}(t:t+K-1)$. The following lemma captures this gap. Define the tail-terms as

$$\phi^{(\tau)}(t) \triangleq -\vec{W}^{\text{T}}\vec{X}^{\text{RMHC}^{(\tau)}}(t), \tag{19a}$$

$$\psi^{(\tau)}(t) \triangleq [\vec{W}^{\text{T}} - \vec{C}^{\text{T}}(t+1)]^{+}\vec{X}^{\text{RMHC}^{(\tau)}}(t). \tag{19b}$$

Lemma 3.2. *(Per-round gap between the primal and dual costs of RMHC$^{(\tau)}$) For each $\tau \in [0, K-1]$ and for any $t = \tau + Ku$ where $u = -1, 0, \ldots, \lfloor \frac{\mathcal{T}}{K} \rfloor$, we have*

$$\begin{aligned} Cost^{RMHC^{(\tau)}}(t:t+K-1) &\le D^{RMHC^{(\tau)}}(t:t+K-1) \\ &+ \phi^{(\tau)}(t-1) + \psi^{(\tau)}(t+K-1), \ \textit{for all } \vec{A}(t:t+K-1). \end{aligned} \tag{20}$$

Lemma 3.2 shows that the primal cost incurred by $\text{RMHC}^{(\tau)}$ in each round is upper-bounded by the online dual cost plus two additional tail-terms. Intuitively, the first tail-term $\psi^{(\tau)}(t+K-1)$ is simply the regularization term. The other tail-term $\phi^{(\tau)}(t-1)$ is due to the initial condition $\vec{X}^{\text{RMHC}^{(\tau)}}(t-1)$, which appears in constraint (16c). Then, $\vec{X}^{\text{RMHC}^{(\tau)}}(t-1)$ will enter the Lagrangian, which leads to $\phi^{(\tau)}(t-1)$. The rest follows from the strong duality theorem [2, p.260]. Please see Appendix C for the proof of Lemma 3.2.

Step 2 (Construct a feasible offline dual solution): Similar to $\text{FHC}^{(\tau)}$, we now wish to concatenate multiple rounds of $\text{RMHC}^{(\tau)}$ together. We then compare the total primal cost $\text{Cost}^{\text{RMHC}^{(\tau)}}(1:\mathcal{T})$ of $\text{RMHC}^{(\tau)}$ with its total online dual cost $D^{\text{RMHC}^{(\tau)}}(1:\mathcal{T})$, which will be no smaller than the optimal offline dual $D^{\text{OPT}}(1:\mathcal{T})$ if the online dual variables are feasible for the offline problem. However, we face a new difficulty. That is, simply concatenating multiple rounds of the same version of $\text{RMHC}^{(\tau)}$, i.e., with the same τ, will not result into feasible dual variables for the offline dual problem (14). This is because, according to Eq.(14), for any time $t = \tau + Ku$, the dual variables should satisfy the constraint $\vec{C}(t+K-1) - \boldsymbol{B}_1^{\text{T}}\vec{\beta}(t+K-1) + \vec{\theta}(t+K) - \vec{\theta}(t+K-1) \ge 0$. However, the dual variables at time $t+K-1$ and time $t+K$ are obtained from the duals of $Q^{\text{RMHC}^{(\tau)}}(t)$ and $Q^{\text{RMHC}^{(\tau)}}(t+K)$, respectively. Since these two optimization problems are independent, the decisions may not satisfy the above constraint. We note that extending the regularization method of [3] to perfect look-ahead will encounter the same difficulty in constructing feasible dual solutions. Specifically, the method of [3] requires setting $\vec{\theta}(t+1)$ to be a specific function of $\vec{X}(t)$, so that the online dual variables remain feasible for the offline optimization problem. However, with look-ahead, such relation is too restrictive because it limits our capability to choose better decisions based on look-ahead prediction.

To resolve the above difficulty, we introduce a new way of "re-stitching" together different rounds of $\text{RMHC}^{(\tau)}$. Let $\text{RMHC}^{(\tau)}(t)$ denote the round of $\text{RMHC}^{(\tau)}$ from time t to time $t+K-1$, where $t = \tau + Ku$ for some integer u. Our key idea is to stitch $\text{RMHC}^{(\tau)}(t)$ together with $\text{RMHC}^{((\tau+1) \bmod K)}(t+K+1)$, but not $\text{RMHC}^{(\tau)}(t+K)$. Note that there is a gap at time $t+K$ between $\text{RMHC}^{(\tau)}(t)$ and $\text{RMHC}^{((\tau+1) \bmod K)}(t+K+1)$. We then add carefully-chosen dual variables at time $t+K$ so that the entire sequence of dual variables is feasible to the offline dual problem (14).

Specifically, before "re-stitching", we could organize all rounds of $D^{\text{RMHC}^{(\tau)}}(t : t+K-1)$, $t = \tau + Ku$, into a matrix form G^{old} in Eq.(21). (Here, for any time index out of the range of $[1, \mathcal{T}]$, by convention we can simply set the corresponding terms of the online dual components to 0.) However, as we discussed above, the dual variables from each row of G^{old} may not form a feasible solution to the offline dual optimization problem. With the "re-stitching" idea, we instead re-organize them into a new matrix G^{new} in Eq.(22). Here, for each row $i = 1, 2, \ldots, K+1$, we let $\tau_i(u) = (i + (K+1)u) \bmod K$, $u = -1, 0, \ldots, \mathcal{U}$, where $\mathcal{U} = \lfloor \frac{\mathcal{T}}{K+1} \rfloor$. Note that the non-zero terms in G^{new} has a one-to-one correspondence to the terms in G^{old}. Each row of G^{new} corresponds to components of G^{old} along a diagonal direction. There is a new row at the bottom of G^{new} because the corresponding terms are skipped in row 1 to K. For each 0 in G^{new}, we assign the dual variables as follows. For the i-th row, 0 in G^{new} corresponds to time slots $i + (K+1)u + K$, where $u = -1, 0, \ldots, \lfloor \frac{\mathcal{T}}{K+1} \rfloor$. We then use

$$\begin{cases} \tilde{\vec{\beta}}(i + (K+1)u + K) = \vec{0}, \\ \tilde{\vec{\theta}}(i + (K+1)u + K) = [\vec{W} - \vec{C}(i + (K+1)u + K)]^+, \end{cases} \tag{23}$$

for each time $i + (K+1)u + K$ between $D^{\text{RMHC}^{(\tau_i(u))}}(i + (K+1)u : i + (K+1)u + K - 1)$ and $D^{\text{RMHC}^{(\tau_i(u+1))}}(i + (K+1)(u+1) : i + (K+1)(u+1) + K - 1)$. Note that, since $\tilde{\vec{\beta}}(i + (K+1)u + K) = \vec{0}$, the corresponding dual cost added to the total online dual cost is zero, which is why we simply write 0 in G^{new}.

Let $G_{i,j}$ be the entry on the i-th row, j-th column of a matrix G. $G_{i,j}$ may be either a dual cost or a 0 term. Then, Lemma 3.3 below shows two important properties of G^{new}.

Lemma 3.3. *(Properties of G^{new}) (1) The sum of all the entries in the original matrix G^{old} is upper-bounded by that in the new matrix G^{new}, i.e.,*

$$\sum_{i=1}^{K} \sum_{j=1}^{\lfloor \frac{\mathcal{T}}{K} \rfloor + 2} G^{old}_{i,j} \leq \sum_{i=1}^{K+1} \sum_{j=1}^{2\lfloor \frac{\mathcal{T}}{K+1} \rfloor + 3} G^{new}_{i,j}. \tag{24}$$

(2) Online dual variables from each row of G^{new} form a feasible solution to the offline dual optimization problem (14).

Part (1) of Lemma 3.3 holds because we are only rearranging the components of G^{old} into G^{new} and adding 0 terms. The key of checking part (2) of Lemma 3.3 is to verify constraint (14c) at time $t+K-1$ and $t+K$, where $t = i + (K+1)u$. Indeed, we can show that $\vec{\theta}(i + (K+1)u + K + 1) = \vec{W}$. Combining with (23), we can then verify, for row i, the feasibility constraint. Please see Appendix D for the proof of Lemma 3.3.

Remark 4. Satisfying constraint (14c) is also the main reason why we choose the regularization term in the form of (9). Specifically, suppose that we replace the regularization term in (9) with a more general form of $-R(t) \cdot \vec{X}(t)$. Focus on a 0 term in row i of G^{new}. Note that this 0 term corresponds to time slot $i + (K+1)u + K$ for some integer u, and is between $\text{RMHC}^{(\tau_i(u))}$ and $\text{RMHC}^{(\tau_i(u+1))}$. In the rest of this remark, we let $t = i + (K+1)u$. First, by writing the equivalent formulation of (9)

(similar to (16) but using the general regularization term of $-R(t)\cdot\vec{X}(t)$) and its KKT conditions at time $t+K-1$, we must have,

$$\vec{C}(t+K-1)-\boldsymbol{B}_1^{\mathrm{T}}\vec{\beta}^{\mathrm{RMHC}^{(\tau_i(u))}}(t+K-1)+\vec{\theta}^{\mathrm{RMHC}^{(\tau_i(u))}}(t+K-1)-R(t+K-1)\geq 0. \tag{25}$$

Thus, in order to satisfy constraint (14c) at time $t+K-1$, it is sufficient to set

$$\tilde{\vec{\theta}}(t+K)\leq R(t+K-1). \tag{26}$$

Second, note that $\mathrm{RMHC}^{(\tau_i(u+1))}$ optimizes over time $t+K+1$ to time $t+2K$. In order to satisfy constraint (14c) at time $t+K$, we need the online dual variables to satisfy

$$\vec{C}(t+K)-\boldsymbol{B}_1^{\mathrm{T}}\tilde{\vec{\beta}}(t+K)+\tilde{\vec{\theta}}(t+K)-\vec{\theta}^{\mathrm{RMHC}^{(\tau_i(u+1))}}(t+K+1)\geq 0. \tag{27}$$

Thus, by choosing $\tilde{\vec{\beta}}(t+K)=\vec{0}$ as in (23), it is sufficient to set

$$\tilde{\vec{\theta}}(t+K)\geq\vec{\theta}^{\mathrm{RMHC}^{(\tau_i(u+1))}}(t+K+1)-\vec{C}(t+K)=\vec{W}-\vec{C}(t+K), \tag{28}$$

where in the last step we have used $\vec{\theta}^{\mathrm{RMHC}^{(\tau_i(u+1))}}(t+K+1)=\vec{W}$ (see the detailed proof in Appendix D). Third, from the definition of the dual variables, we have

$$\vec{W}\geq\tilde{\vec{\theta}}(t+K)\geq 0. \tag{29}$$

Finally, combining Eqs.(26), (28) and (29), it is sufficient to choose $R(\cdot)$ such that

$$\vec{W}\geq R(t+K-1)\geq[\vec{W}-\vec{C}(t+K)]^{+}. \tag{30}$$

Our regularization term corresponds to $R(t+K-1)=[\vec{W}^{\mathrm{T}}-\vec{C}^{\mathrm{T}}(t+K)]^{+}$, because it has the smallest impact on the objective value of Eq.(9). Note that we can also choose $R(t+K-1)=\vec{W}$, which is the case in Sec. 3.4.

Step 3: We next show in Lemma 3.4 that the sum of the tail-terms from the same version τ of $\mathrm{RMHC}^{(\tau)}$ over all rounds is non-positive.

Lemma 3.4. *(The total contribution of the tail-terms) For each $\tau\in[0,K-1]$, we have*

$$\sum_{u=-1}^{\lfloor\frac{\mathcal{T}}{K}\rfloor}\left\{\phi^{(\tau)}(\tau+Ku-1)+\psi^{(\tau)}(\tau+Ku+K-1)\right\}\leq 0. \tag{31}$$

This is mainly because $[\vec{W}^{\mathrm{T}}-\vec{C}^{\mathrm{T}}(\tau+Ku+K)]^{+}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(\tau+Ku+K-1)\leq\vec{W}^{\mathrm{T}}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(\tau+Ku+K-1)$ and we can then cancel these terms. Please see Appendix E for the proof of Lemma 3.4.

Now we are ready to provide a complete proof for Theorem 3.1.

Proof. (Proof of Theorem 3.1) First, from the cost definition in Eq.(4), we know that the total cost of ARMHC is

$$\mathrm{Cost}^{\mathrm{ARMHC}}(1:\mathcal{T})=\sum_{t=1}^{\mathcal{T}}\vec{C}^{\mathrm{T}}(t)\vec{X}^{\mathrm{ARMHC}}(t)+\sum_{t=1}^{\mathcal{T}}\vec{W}^{\mathrm{T}}[\vec{X}^{\mathrm{ARMHC}}(t)-\vec{X}^{\mathrm{ARMHC}}(t-1)]^{+}. \tag{32}$$

Then, from Eq.(11) that ARMHC takes the average of K versions of RMHC decisions, we have

$$\begin{aligned}\mathrm{Cost}^{\mathrm{ARMHC}}(1:\mathcal{T})=&\sum_{t=1}^{\mathcal{T}}\left\{\vec{C}^{\mathrm{T}}(t)\cdot\left[\frac{1}{K}\sum_{\tau=0}^{K-1}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t)\right]\right\}\\&+\sum_{t=1}^{\mathcal{T}}\left\{\vec{W}^{\mathrm{T}}\cdot\left[\frac{1}{K}\sum_{\tau=0}^{K-1}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t)-\frac{1}{K}\sum_{\tau=0}^{K-1}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t-1)\right]^{+}\right\}.\end{aligned} \tag{33}$$

Due to the convexity of $[\cdot]^+$ and Jensen's Inequality, we have

$$\begin{aligned}\mathrm{Cost}^{\mathrm{ARMHC}}(1:\mathcal{T}) \leq & \frac{1}{K}\sum_{t=1}^{\mathcal{T}}\sum_{\tau=0}^{K-1}\vec{C}^{\mathrm{T}}(t)\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t) \\ & + \frac{1}{K}\sum_{t=1}^{\mathcal{T}}\sum_{\tau=0}^{K-1}\vec{W}^{\mathrm{T}}[\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t) - \vec{X}^{\mathrm{RMHC}^{(\tau)}}(t-1)]^+.\end{aligned} \tag{34}$$

Next, we change the order of the summations over t and τ. Meanwhile, we put together all rounds of $\mathrm{RMHC}^{(\tau)}$ from the same row of G^{old}. We can then get

$$\mathrm{Cost}^{\mathrm{ARMHC}}(1:\mathcal{T}) \leq \frac{1}{K}\sum_{\tau=0}^{K-1}\sum_{u=-1}^{\lfloor\frac{\mathcal{T}}{K}\rfloor}\mathrm{Cost}^{\mathrm{RMHC}^{(\tau)}}(\tau+Ku:\tau+Ku+K-1). \tag{35}$$

According to Lemma 3.2, we have

$$\begin{aligned}&\mathrm{Cost}^{\mathrm{ARMHC}}(1:\mathcal{T}) \\ &\leq \frac{1}{K}\sum_{\tau=0}^{K-1}\sum_{u=-1}^{\lfloor\frac{\mathcal{T}}{K}\rfloor}\left\{D^{\mathrm{RMHC}^{(\tau)}}(\tau+Ku:\tau+Ku+K-1) + \phi^{(\tau)}(\tau+Ku-1) + \psi^{(\tau)}(\tau+Ku+K-1)\right\} \\ &\leq \frac{1}{K}\sum_{\tau=0}^{K-1}\sum_{u=-1}^{\lfloor\frac{\mathcal{T}}{K}\rfloor}D^{\mathrm{RMHC}^{(\tau)}}(\tau+Ku:\tau+Ku+K-1),\end{aligned} \tag{36}$$

where we have used Lemma 3.4 in the last step. Then, applying the re-stitching idea, we have,

$$\begin{aligned}\sum_{\tau=0}^{K-1}\sum_{u=-1}^{\lfloor\frac{\mathcal{T}}{K}\rfloor}D^{\mathrm{RMHC}^{(\tau)}}(\tau+Ku:\tau+Ku+K-1) &= \sum_{i=1}^{K}\sum_{j=1}^{\lfloor\frac{\mathcal{T}}{K}\rfloor+2}G^{\mathrm{old}}_{i,j} \\ &\leq \sum_{i=1}^{K+1}\sum_{j=1}^{2\lfloor\frac{\mathcal{T}}{K+1}\rfloor+3}G^{\mathrm{new}}_{i,j} \leq (K+1)D^{\mathrm{OPT}}(1:\mathcal{T}),\end{aligned} \tag{37}$$

where the first inequality follows from part (1) of Lemma 3.3, and the second inequality follows from part (2) of Lemma 3.3 and that the dual objective value of any feasible dual variables must be no larger than the optimal offline dual cost $D^{\mathrm{OPT}}(1:\mathcal{T})$. Combining Eqs.(36), (37) and (15), we have

$$\begin{aligned}\mathrm{Cost}^{\mathrm{ARMHC}}(1:\mathcal{T}) \leq & \frac{1}{K}\cdot(K+1)D^{\mathrm{OPT}}(1:\mathcal{T}) \\ = & \frac{K+1}{K}\cdot\mathrm{Cost}^{\mathrm{OPT}}(1:\mathcal{T}),\ \text{for all } \mathcal{T} \text{ and } \vec{A}(1:\mathcal{T}).\end{aligned} \tag{38}$$

□

3.4 Convex Service Costs

The above results assume that the service cost is linear in $\vec{X}(t)$. For the case with perfect look-ahead, our results can be generalized to convex service costs, with a minor change of ARMHC. Specifically, here the service cost at time t is given by $h_t(\vec{X}(t))$, where $h_t(\cdot)$ is a non-negative convex function.

In Algorithm 2, instead of solving the optimization problem (9), now ARMHC solves (39) below,

$$\begin{aligned} \min_{\vec{X}(t:t+K-1)} & \left\{ \sum_{s=t}^{t+K-1} \left[h_s(\vec{X}(s)) + \vec{W}^{\mathrm{T}}[\vec{X}(s) - \vec{X}(s-1)]^+ \right] - \vec{W}^{\mathrm{T}}\vec{X}(t+K-1) \right\} \\ \text{sub. to: } & \boldsymbol{B}_1\vec{X}(s) \geq \boldsymbol{B}_2\vec{A}(s), \text{ for all time } s \in [t, t+K-1], \\ & \vec{X}(s) \geq 0, \text{ for all time } s \in [t, t+K-1], \end{aligned} \tag{39}$$

Note that the regularization term is changed to $-\vec{W}^{\mathrm{T}}\vec{X}(t+K-1)$ because we no longer have a fixed $\vec{C}(t)$. We then have Theorem 3.5 as follows.

Theorem 3.5. *When the online convex optimization problem formulated in Sec. 2.1 uses convex service costs $h_t(\vec{X}(t))$, the Averaging Regularized Moving Horizon Control (ARMHC) algorithm described above is $(\frac{K+1}{K})$-competitive.*

Please see Appendix F for the proof of Theorem 3.5.

4 THE IMPERFECT LOOK-AHEAD CASE

In the previous section, we have assumed that the future information in the look-ahead window of size K is perfect. In practice, such an assumption is often too strong. Although one can predict the future input in the look-ahead window, such predictions are usually not precise. In this section, we will develop competitive online algorithms for such cases with imperfect look-ahead.

Recall our model of imperfect look-ahead in Sec. 2.2, where the decision maker is given the current input along with a predicted input trajectory $\vec{A}_t^{\text{pred}}(t+1:t+K)$ in the look-ahead window. Moreover, she knows the upper and lower bounds defined by $\vec{f}^{\text{up}}(1:K)$ and $\vec{f}^{\text{low}}(1:K)$ as in Eq.(3). Our proposed algorithms will utilize such knowledge of the prediction accuracy in the look-ahead window to achieve low competitive ratios.

Note that the Committed Horizon Control (CHC) in [8] also aims to deal with imperfect look-ahead. However, the results there only provide guarantees on the competitive difference of CHC, not its competitive ratio. Further, the results there assume no hard constraints that depend on inputs. Thus, it remains an open problem to develop online algorithms with low competitive ratios under imperfect look-ahead and such hard constraints.

4.1 Dealing with Uncertain Hard Constraints under Imperfect Look-Ahead

As we mentioned at the end of Sec. 2.2, combining imperfect look-ahead with uncertain hard constraint creates new challenges. Note that in the perfect look-ahead case, the predicted inputs in the look-ahead window is precise. Hence, the decisions solved by RMHC$^{(\tau)}$ in Eq.(9) always satisfy the real constraint (1) in the future, so do the ARMHC decisions when an average is taken over different versions of RMHC$^{(\tau)}$. However, in the imperfect look-ahead case, due to the prediction error, the real inputs in the look-ahead window might be larger than the predicted values. If we still use the predicted input in the constraints in Eq.(9), the decisions obtained by ARMHC may not satisfy the hard constraints with real inputs.

A simple remedy is to use the upper bound of the future inputs in the optimization problem (9). This becomes RMHC-IL (Regularized Moving Horizon Control-Imperfect Look-ahead), which can be described as follows. At each time t, each version τ of RMHC-IL$^{(\tau)}$ simply replaces (9b) and (10b) of time $t+1$ to time $t+K-1$ by

$$\boldsymbol{B}_1\vec{X}(s) \geq \boldsymbol{B}_2\overline{\vec{A}}(s), \text{ for all time } s \in [t+1, t+K-1], \tag{40}$$

where $\overline{\vec{A}}(s) = \{\overline{a}_m(s),\ \text{for all } m \in [1, M]\}$ and $\overline{a}_m(s) = a^{\text{pred}}_{t,m}(s) \cdot f^{\text{up}}_m(s-t)$ is the upper bound of the future inputs. Then, by taking an average across multiple versions of RMHC-IL$^{(\tau)}$, we have a revised version of ARMHC for the imperfect look-ahead case. We call it Averaging Regularized Moving Horizon Control-Imperfect Look-ahead (ARMHC-IL). Theorem 4.1 below provides the competitive ratio of the ARMHC-IL.

Theorem 4.1. *The competitive ratio of the Averaging Regularized Moving Horizon Control-Imperfect Look-ahead (ARMHC-IL) algorithm is* $\frac{K+1}{K} \cdot \max_{\{i \in [0, K-1], m \in [1, M]\}} \frac{f^{up}_m(i)}{f^{low}_m(i)}$.

We omit the proof of Theorem 4.1 because it can be viewed as a special case of the more general result in Theorem 4.2 later (by letting $\rho(i) = 0$ for all $i \in [0, K-1]$ there). To the best of our knowledge, ARMHC-IL is the first algorithm to achieve a theoretically-guaranteed competitive ratio under imperfect look-ahead for online convex optimization (OCO) problem with uncertain hard constraints. The competitive ratio in Theorem 4.1 has two parts. The first part, $\frac{K+1}{K}$, is from the performance of ARMHC shown in Theorem 3.1 in Sec. 3.3. The second part is based on the gap between the upper and lower bounds. Thus, it captures the impact of the prediction error. If there is no prediction error, the second part is 1. We then recover the results with perfect look-ahead.

However, the competitive ratio of ARMHC-IL is still highly sensitive to the bounds on the prediction accuracy. In particular, the competitive ratio of ARMHC-IL is dominated by the worst prediction quality. Intuitively, this dependency of ARMHC-IL on the worst-case prediction accuracy is because ARMHC-IL treats all time slots uniformly. Knowing the prediction accuracy of each future time slot, we could have treated the time slots and different rounds of RMHC$^{(\tau)}$ differently, according to how reliable the prediction is. The following Weighting Regularized Moving Horizon Control (WRMHC) assigns different weights to inputs and decisions, which can achieve a lower competitive ratio than ARMHC-IL and has lower dependency on the worst-case prediction accuracy.

4.2 Weighting Regularized Moving Horizon Control

Different from AFHC or ARMHC, Weighting Regularized Moving Horizon Control (WRMHC) takes the *weighted* average of the decisions from multiple rounds of a subroutine, called Generalized Regularized Moving Horizon Control (GRMHC). In view of the different prediction quality at different times, GRMHC adds weights $\rho(0 : K-1) = \{\rho(i), \text{for all } i \in [0, K-1]\}$ to different terms in the optimization problem (9). Moreover, GRMHC may not use the entire look-ahead window of size K if the prediction quality in the later part of the look-ahead window is very poor. Specifically, let the size of the committed look-ahead window be k, $1 \le k \le K$. Then, GRMHC may only use the predicted inputs in the first k time-slots of the look-ahead window. We will show how to choose the optimal value for k and $\rho(0 : k-1)$ at the end of this section. Let τ be an integer from the set $[0, k-1]$. At each time $t = \tau + ku$, $u = -1, 0, \ldots, \lfloor \frac{\mathcal{T}}{k} \rfloor$, the τ-th version of GRMHC, denoted by GRMHC$^{(\tau)}$, is given the current input $\vec{A}(t)$ and the predicted inputs $\vec{A}^{\text{pred}}_t(t+1 : t+k)$. Then, GRMHC$^{(\tau)}$ calculates the solution to the following optimization problem $Q^{\text{GRMHC}^{(\tau)}}(t)$ at each time $t = \tau + ku$,

$$\min_{\vec{X}(t:t+k-1)} \left\{ \sum_{s=t}^{t+k-1} \rho(s-t)\vec{C}^{\mathrm{T}}(s)\vec{X}(s) + \vec{W}^{\mathrm{T}} \left[\rho(0)\vec{X}(t) - \rho(k-1)\vec{X}(t-1)\right]^{+} \right.$$
$$+ \sum_{s=t+1}^{t+k-1} \vec{W}^{\mathrm{T}} \left[\rho(s-t)\vec{X}(s) - \rho(s-t-1)\vec{X}(s-1)\right]^{+}$$
$$\left. - \left[\vec{W}^{\mathrm{T}} - \vec{C}^{\mathrm{T}}(t+k)\right]^{+} \cdot \rho(k-1)\vec{X}(t+k-1) \right\} \tag{41a}$$

$$\text{sub. to: } \boldsymbol{B}_1\vec{X}(t) \geq \boldsymbol{B}_2\vec{A}(t), \tag{41b}$$

$$\boldsymbol{B}_1\vec{X}(s) \geq \boldsymbol{B}_2\overline{\vec{A}}(s), \text{ for all time } s \in [t+1, t+k-1], \tag{41c}$$

$$\vec{X}(s) \geq 0, \text{ for all time } s \in [t, t+k-1]. \tag{41d}$$

Note that a weight $\rho(s-t)$ is assigned to each time slot s in the objective of (41). Intuitively, with a smaller $\rho(s-t)$, the cost of the corresponding time s will contribute less to the total cost. In this way, the input of the corresponding time will have less influence on the decisions. Since GRMHC$^{(\tau)}$ only commits to the first k slots of the entire look-ahead window, the initial value $\vec{X}(t-1)$ for $Q^{\mathrm{GRMHC}^{(\tau)}}(t)$ is from GRMHC$^{(\tau)}$ computed at an earlier time of $t-k$, when $Q^{\mathrm{GRMHC}^{(\tau)}}(t-k)$ was solved. Let $\tilde{\tau}(t,i) \triangleq (t-i) \bmod k$ for any $i \in [0, k-1]$. Then, at each time t, WRMHC takes the *weighted* average of $\vec{X}^{\mathrm{GRMHC}^{(\tau)}}(t)$ for all $\tau \in [0, k-1]$,

$$\vec{X}^{\mathrm{WRMHC}}(t) = \frac{\sum_{i=0}^{k-1} \rho(i)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,i))}}(t)}{\sum_{i=0}^{k-1} \rho(i)}. \tag{42}$$

Similar to RMHC, if $t + K > \mathcal{T}$, GRMHC solves $\tilde{Q}^{\mathrm{GRMHC}^{(\tau)}}(t)$ in Eq.(43) instead of $Q^{\mathrm{GRMHC}^{(\tau)}}(t)$ in Eq.(41). The details of WRMHC are given in Algorithm 3.

In summary, comparing WRMHC with ARMHC-IL, we can see two important differences. The first one is the different weights assigned to each $\vec{X}(s)$ in the objective of (41). These weights represent different importance of the cost-terms due to different prediction quality. The second one is the different weights in the averaging in Eq.(42), which again reflect different importance due to different prediction quality. Theorem 4.2 below gives the competitive ratio of WRMHC.

Theorem 4.2. *The competitive ratio of the Weighting Regularized Moving Horizon Control (WRMHC) algorithm with the committed look-ahead window of size k and weights $\rho(0:k-1)$ is*

$$\frac{k+1}{\sum_{i=0}^{k-1} \rho(i)} \max_{\{i\in[0,k-1], m\in[1,M]\}} \frac{f_m^{up}(i)}{f_m^{low}(i)}\rho(i). \tag{44}$$

In other words, the competitive ratio of WRMHC contains two parts. The first part, $\frac{(k+1)}{\sum_{i=0}^{k-1}\rho(i)}$, is still similar to that of ARMHC in Theorem 3.1 for the perfect look-ahead case. The only difference is that the denominator of this term is the sum of k different weights. The second part, $\max_{\{i\in[0,k-1], m\in[1,M]\}} \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)}\rho(i)$, is not only based on the gap between the upper and lower bounds,

Algorithm 3 Weighting Regularized Moving Horizon Control (WRMHC) with committed look-ahead window of size k

Input: $\vec{A}(1:\mathcal{T})$, K, $\vec{f}^{\text{up}}(1:K)$ and $\vec{f}^{\text{low}}(1:K)$.
Output: $\vec{X}^{\text{WRMHC}}(1:\mathcal{T})$.
FOR $t = -k+2:\mathcal{T}$
Step 1: $\tau \leftarrow t \bmod k$, and then $\vec{X}(t-1) \leftarrow \vec{X}^{\text{GRMHC}^{(\tau)}}(t-1)$.
Step 2: Given the current input $\vec{A}(t)$ and predicted inputs $\vec{A}_t^{\text{pred}}(t+1:t+K)$ in the committed look-ahead window,
if $t+k \le \mathcal{T}$ **then**
solve problem (41) to get $\vec{X}^{\text{GRMHC}^{(\tau)}}(t:t+k-1)$,
else
solve problem (43) to get $\vec{X}^{\text{GRMHC}^{(\tau)}}(t:\mathcal{T})$,

$$\begin{aligned}\min_{\vec{X}(t:\mathcal{T})} \Bigg\{ & \sum_{s=t}^{\mathcal{T}} \rho(s-t)\vec{C}^{\text{T}}(s)\vec{X}(s) + \vec{W}^{\text{T}}\left[\rho(0)\vec{X}(t) - \rho(k-1)\vec{X}(t-1)\right]^{+} \\ & + \sum_{s=t+1}^{\mathcal{T}} \vec{W}^{\text{T}}\left[\rho(s-t)\vec{X}(s) - \rho(s-t-1)\vec{X}(s-1)\right]^{+} \Bigg\} \\ \text{sub. to: } & \boldsymbol{B}_1\vec{X}(t) \ge \boldsymbol{B}_2\vec{A}(t), \\ & \boldsymbol{B}_1\vec{X}(s) \ge \boldsymbol{B}_2\overline{\vec{A}}(s), \text{ for all time } s \in [t+1, \mathcal{T}], \\ & \vec{X}(s) \ge 0, \text{ for all time } s \in [t, \mathcal{T}].\end{aligned} \tag{43}$$

end if
Step 3: Use Eq.(42) to get the final decision $\vec{X}^{\text{WRMHC}}(t)$.
END

but also based on the weights. This result then allows us to set appropriate weights to offset the influence from poor predictions. Please see Appendix G for the complete proof of Theorem 4.2.

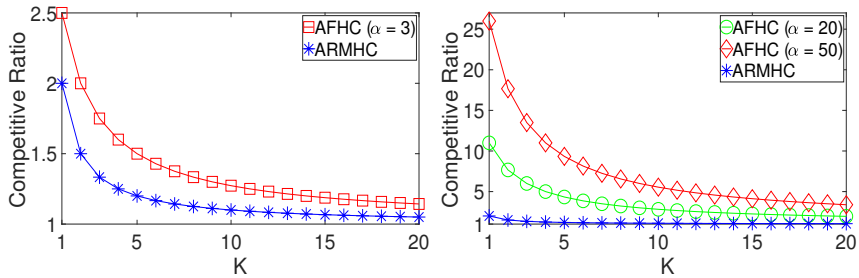
Based on Theorem 4.2, we now discuss how to set the weights and choose the optimal size k^* of the committed look-ahead window.

Lemma 4.3. *For any $k \in [1, K]$, the optimal weight $\rho(0:k-1)$ for minimizing the competitive ratio in Eq.(44) is given by*

$$\begin{cases} \rho^*(0) = 1, \\ \rho^*(i) = \min\limits_{m\in[1,M]} \frac{f_m^{low}(i)}{f_m^{up}(i)}, \text{ for all } i \in [1, k-1]. \end{cases} \tag{45}$$

See Appendix H for the proof of Lemma 4.3. Then, the competitive ratio of WRMHC becomes $\frac{k+1}{1+\sum_{i=1}^{k-1}\min_{m\in[1,M]}\frac{f_m^{\text{low}}(i)}{f_m^{\text{up}}(i)}}$. Finally, we can obtain the optimal size of the committed look-ahead window as

$$k^* = \arg\min_{1\le k\le K} \frac{k+1}{1+\sum_{i=1}^{k-1}\min_{m\in[1,M]}\frac{f_m^{\text{low}}(i)}{f_m^{\text{up}}(i)}}. \tag{46}$$



(a) $\mathrm{CR}^{\mathrm{AFHC}}(\alpha = 3)$; $\mathrm{CR}^{\mathrm{ARMHC}}$. (b) $\mathrm{CR}^{\mathrm{AFHC}}(\alpha = 20, 50)$; $\mathrm{CR}^{\mathrm{ARMHC}}$.

Fig. 3. Theoretical competitive ratios of AFHC(α) for various values of $\alpha = 3, 20, 50$ and ARMHC. ($\alpha = \max_{\{n\in[1,N], t\in[1,\mathcal{T}]\}} \frac{w_n}{c_n(t)}$)

Theorem 4.4. *Using the weights in Eq.(45) and the optimal size of committed look-ahead window k^* given in Eq.(46), the competitive ratio of WRMHC* is*

$$\mathrm{CR}^{WRMHC^*} = \frac{k^* + 1}{1 + \sum_{i=1}^{k^*-1} \min_{m\in[1,M]} \frac{f_m^{low}(i)}{f_m^{up}(i)}}. \tag{47}$$

Compared with the competitive ratio of ARMHC-IL in Theorem 4.1, the competitive ratio in Theorem 4.4 is less sensitive to the worst-case prediction $\max_{i\in[0,K-1], m\in[1,M]} \left\{\frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)}\right\}$. For example, the competitive ratio in Eq.(47) will never be larger than $k^* + 1$, while the competitive ratio in Theorem 4.1 can potentially be much larger than $k^* + 1$. This insensitivity is because WRMHC assigns a much smaller weight $\rho(i) = \min_{m\in[1,M]} \frac{f_m^{\mathrm{low}}(i)}{f_m^{\mathrm{up}}(i)}$ for the worst-case prediction.

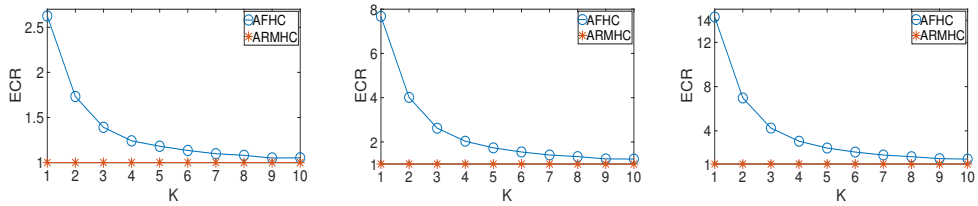
Finally, from Algorithm 3 we notice that when the look-ahead is perfect, i.e., $f_m^{\mathrm{up}}(i) \equiv 1$ and $f_m^{\mathrm{low}}(i) \equiv 1$ for all $m \in [1, M]$ and $i \in [1, K]$, WRMHC* will choose the weight $\rho(i) = 1$ for all $i \in [0, K-1]$ and the size of the committed look-ahead window becomes $k^* = K$. Then, WRMHC* becomes ARMHC as in the perfect look-ahead case. Hence, WRMHC* can also be regarded as a generalized version of ARMHC that works for both perfect and imperfect look-ahead.

5 NUMERICAL RESULTS

In this section, we compare the theoretical competitive ratios of our proposed algorithms and the existing algorithms under perfect and imperfect look-ahead. Further, using traces either from Counter-example 1 or from the HP trace in [10], we conduct numerical simulations comparing the empirical performance of ARMHC and AFHC for perfect look-ahead, and that of WRMHC and CHC for imperfect look-ahead. Throughout, CR^{π} is the competitive ratio of an online algorithm π.

5.1 Perfect Look-Ahead

In this section, we focus on the case with perfect look-ahead. We first compare the competitive ratio of ARMHC with that of AFHC. AFHC has been proposed in [20] for perfect look-ahead. Its competitive ratio is shown to be $1 + \frac{\alpha}{K+1}$, where $\alpha = \max_{\{n\in[1,N], t\in[1,\mathcal{T}]\}} \frac{w_n}{c_n(t)}$. In contrast, the competitive ratio of our proposed ARMHC is $\frac{K+1}{K}$. These competitive ratios are plotted in Fig. 3.



(a) ECR: $\vec{W} \geq 3\vec{C}(t)$. (b) ECR: $\vec{W} \geq 20\vec{C}(t)$. (c) ECR: $\vec{W} \geq 50\vec{C}(t)$.

Fig. 4. Empirical competitive ratios of AFHC and ARMHC for the input in Counter-example 1.

From Fig. 3, we can observe that the competitive ratios of both AFHC and ARMHC decrease as the size of the look-ahead window K increases. However, the competitive ratio of AFHC is much larger than that of ARMHC, especially when K is not very large and α is very large. For example, when $K = 2$, CR^{AFHC} is 7.667 and 17.667 for $\alpha = 20$ and 50, respectively. In contrast, the competitive ratio of our proposed ARMHC is only 1.5. Thus, ARMHC achieves significantly lower competitive ratios for online convex optimization (OCO) problems with perfect look-ahead.

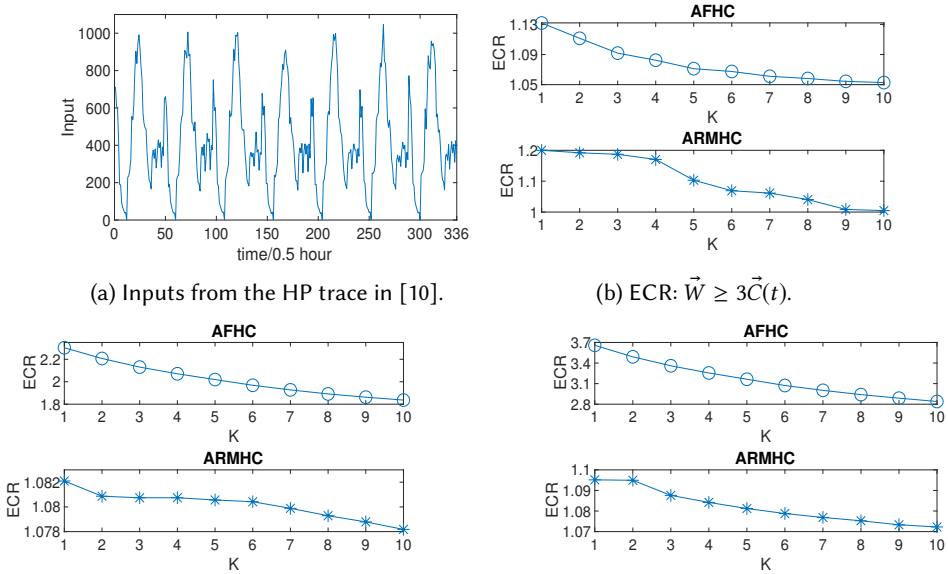
Then, we provide numerical results comparing their empirical performance. We take $\mathcal{T} = 336$. While the switching-cost coefficient $\vec{W}$ is fixed, we allow the service-cost coefficient $\vec{C}(t)$ to change in time. $\vec{C}(t)$ are uniformly generated between 0 and 1, i.e., $\vec{C}(t) \sim U(0, 1)$. For Figs. 4a and 5b, we use $\vec{W} \sim U(3, 4)$, and thus $\vec{W} \geq 3\vec{C}(t)$. For Figs. 4b and 5c, $\vec{W} \sim U(20, 30)$, thus $\vec{W} \geq 20\vec{C}(t)$. For Figs. 4c and 5d, $\vec{W} \sim U(50, 60)$, thus $\vec{W} \geq 50\vec{C}(t)$.

In Fig. 4, we compare the empirical performance of ARMHC with that of AFHC using the input presented in Counter-example 1 in Sec. 3.1. Here, we take $N = 10$, $M = 30$, $L = 5$ and all entries of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are generated under $U(0, 1)$. In the rest of the simulation, we define the empirical competitive ratio (ECR) to be the ratio of total cost of an online algorithm to that of the optimal offline solution for a given input. Fig. 4 shows results for one realization. We observe that ARMHC nearly performs optimally with ECR close to 1. In contrast, AFHC has a large ECR for all three cases of the switching-cost coefficients. The gap is especially large when K is not very large.

In Fig. 5, we compare the empirical performance of ARMHC with that of AFHC using the inputs generated from the HP trace in [10] (re-produced in Fig. 5a). Notice that here $M = 1$. Moreover, we take $N = 3$, $L = 1$ and all entries of $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are 1. We consider one week and the inputs are for every half hour. From Fig. 5, we can observe that, except when both K and $\vec{W}$ are very small, in which case the difference is very small, the cost ratio of ARMHC is much lower than that of AFHC for most other cases.

5.2 Imperfect Look-Ahead

In Fig 6, we compare the theoretical competitive ratios of ARMHC-IL and WRMHC. Here, the bounds are $f_m^{\text{up}}(i) = 1 + \epsilon_{\text{up}} \cdot i$ and $f_m^{\text{low}}(i) = \max\{0, 1 - \epsilon_{\text{low}} \cdot i\}$, for all $i \in [1, K]$, where $\epsilon_{\text{up}} = 0.3$ and $\epsilon_{\text{low}} = 0.025$. First, we can observe that ARMHC-IL is highly sensitive to the prediction error. Specifically, since $\frac{f_m^{\text{up}}(k)}{f_m^{\text{low}}(k)}$ increases as k increases, $\text{CR}^{\text{ARMHC-IL}}$ increases quickly. In contrast, the competitive ratio of WRMHC increases much slower and is always around a value of 2. Second, ARMHC-IL can also choose the optimal size of a committed look-ahead window, i.e., one can get $\text{CR}^{\text{ARMHC-IL}^*} = 2$ when choosing $k^* = 2$. Still, WRMHC* attains a lower competitive ratio, i.e., $\text{CR}^{\text{WRMHC}^*} = 1.7$ when $k^* = 2$.



(a) Inputs from the HP trace in [10].

(b) ECR: $\vec{W} \geq 3\vec{C}(t)$.

(c) ECR: $\vec{W} \geq 20\vec{C}(t)$.

(d) ECR: $\vec{W} \geq 50\vec{C}(t)$.

Fig. 5. Empirical competitive ratios of AFHC and ARMHC for the trace in Fig. 5a.

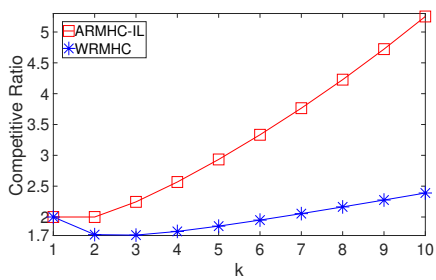


Fig. 6. Theoretical competitive ratios of ARMHC-IL and WRMHC.

Then, we provide numerical results for the empirical performance of WRMHC and CHC under imperfect look-ahead. In all the simulations, we take $\mathcal{T} = 336$. The service-cost coefficient $\vec{C}(t)$ and the switching-cost coefficient $\vec{W}$ are generated randomly in advanced. While $\vec{W}$ is fixed, we allow $\vec{C}(t)$ to change in time. $\vec{C}(t)$ are uniformly generated in the continuous interval from 0 to 1, i.e., $\vec{C}(t) \sim U(0, 1)$. For $\vec{W} \geq 3\vec{C}(t)$, $\vec{W} \geq 20\vec{C}(t)$ and $\vec{W} \geq 50\vec{C}(t)$, we generate $\vec{W}$ the same way as in the numerical analysis for perfect look-ahead in Sec 5.1. For each look-ahead window, we take the trace in Fig. 5a as the real input. Notice that, as for Fig. 6, the bounds are $f_m^{\text{up}}(i) = 1 + \epsilon_{\text{up}} \cdot i$ and $f_m^{\text{low}}(i) = \max\{0, 1 - \epsilon_{\text{low}} \cdot i\}$, for all $i \in [1, K]$, where $\epsilon_{\text{up}} = 0.3$ and $\epsilon_{\text{low}} = 0.025$. Then we generate imperfect look-ahead by adding noise within the range given by $\vec{f}^{\text{up}}(1 : K)$ and $\vec{f}^{\text{low}}(1 : K)$.

We refer to the version of WRMHC with the optimal choice of k^* as WRMHC*. In Table 1, using the inputs presented in Counter-example 1 in Sec. 3.1, we compare the empirical performance of

	$\vec{W} \geq 3\vec{C}(t)$	$\vec{W} \geq 20\vec{C}(t)$	$\vec{W} \geq 50\vec{C}(t)$
ECR of CHC	2.4057	4.7134	9.7890
ECR of WRMHC*	1.1714	1.1411	1.1772

Table 1. Empirical competitive ratios of CHC and WRMHC* for inputs in Counter-example 1.

	$\vec{W} \geq 3\vec{C}(t)$	$\vec{W} \geq 20\vec{C}(t)$	$\vec{W} \geq 50\vec{C}(t)$
ECR of CHC	1.9204	3.1945	5.4397
ECR of WRMHC*	1.7226	1.2966	1.2161

Table 2. Empirical competitive ratios of CHC and WRMHC* for the trace in Fig. 5a.

WRMHC* with that of CHC. Here, we take $N = 10$, $M = 30$, $L = 5$ and all entries of $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ are generated under $U(0, 1)$. From [8], we know that when the product of the switching-cost coefficient and the size of the decision space is large, CHC will always use the entire look-ahead window and become AFHC. Here, we have scaled the values of inputs to be about 1000 and it is large enough for CHC to become AFHC. From Table 1, we observe that WRMHC* nearly performs optimally with ECR close to 1. In contrast, CHC has a large ECR for all three cases of the switching-cost coefficients. The gap is especially large when k is not very large.

In Table 2, we compare the empirical performance of WRMHC* with CHC using the inputs generated from the HP trace at [10] (re-produced in Fig. 5a). Notice that here $M = 1$. Moreover, we take $N = 3$, $L = 1$ and all entries of $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ are 1. From Table 2, we can observe that the empirical competitive ratio of WRMHC* is much lower than that of CHC for all cases. Moreover, when the switching-cost coefficient increases, CHC will get worse since it will pay more costs for switching, likely due to the "end-of-look-ahead" problem we discussed in Sec. 3.1. In contrast, the ECR of WRMHC* decreases as $\vec{W}$ increases since it will switch less when $\vec{W}$ becomes larger.

6 CONCLUSION

In this paper, we study how to best utilize either perfect or imperfect look-ahead to achieve low competitive ratios in online convex optimization (OCO) problems. For perfect look-ahead, we develop the Averaging Regularized Moving Horizon Control (ARMHC) algorithm, which is $\frac{K+1}{K}$-competitive. This is the first result in the literature that can achieve a parameter-independent competitive ratio for this type of cost-minimization problems with perfect look-ahead. For imperfect look-ahead, we propose the Weighting Regularized Moving Horizon Control (WRMHC) algorithm, whose competitive ratio is less sensitive to the prediction errors. It is the first online algorithm with a theoretically-guaranteed competitive ratio under imperfect look-ahead. Further, our analysis combines online primal-dual analysis with a novel re-stitching idea, which is of independent interest.

Throughout the paper, we have made a number of assumptions on the service costs and switching costs, e.g., the service costs are linear and the switching costs are proportional to the increments of decisions across time. Not all assumptions are essential in all cases. For example, for the perfect look-ahead case, we have relaxed the linear service costs to convex service costs in Sec. 3.4. Further, Theorem 3.1 will also hold under capacity constraints. This is because the entries of both $\boldsymbol{B_1}$ and $\boldsymbol{B_2}$ could be negative, without affecting the proofs of our results for the perfect look-ahead case. In this way, constraint (1) can also represent capacity constraints, e.g., the consumption of the resources should not exceed the available amount of resources in Network Functions Virtualization (NFV) [9, 14, 28]. Similarly, we expect that Theorem 3.1 holds for the case when switching-cost

coefficients $\vec{W}$ change in time, but are otherwise known in a perfect look-ahead window. However, we expect that it will be more difficult to relax these assumptions for the imperfect look-ahead case. For example, when the service cost is convex (and non-linear), or when there are capacity constraints, it becomes more difficult to upper-bound the online cost because the ratio between the true online cost and the cost estimated by RMHC-IL (based on the upper-bound of the future inputs) could be larger than $\max_{\{m\in[1,M]\}} \frac{f_{\text{up}}^m(i)}{f_{\text{low}}^m(i)}$. Further, when future switching-cost coefficient $\vec{W}(t)$ at the end of the look-ahead window is not precisely known, it is also more difficult to construct the regularization term in (9a) and (41a) so that the online dual variables are feasible for the offline dual problem. For future work ,we plan to study other ways to deal with such difficulty. Further, we plan to use more realistic traces to evaluate the proposed online algorithms. Finally, an important open question is to derive lower bounds on the competitive ratios for OCO under either perfect or imperfect look-ahead, which will shed important insight on how tight our competitive ratios are for both cases.

ACKNOWLEDGMENTS

This work has been supported in part by NSF grants ECCS-1509536 and CNS-1717493.

REFERENCES

[1] Nikhil Bansal, Anupam Gupta, Ravishankar Krishnaswamy, Kirk Pruhs, Kevin Schewior, and Cliff Stein. 2015. A 2-Competitive Algorithm for Online Convex Optimization with Switching Costs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 96–109.

[2] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[3] Niv Buchbinder, Shahar Chen, and Joseph (Seffi) Naor. 2014. Competitive Analysis via Regularization. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '14)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 436–444.

[4] Niv Buchbinder, Shahar Chen, Joshep (Seffi) Naor, and Ohad Shamir. 2012. Unified Algorithms for Online Learning and Competitive Analysis. In *Proceedings of the 25th Annual Conference on Learning Theory*. Proceedings of Machine Learning Research, Edinburgh, Scotland, 23:5.1–5.18.

[5] Niv Buchbinder and Joseph (Seffi) Naor. 2009. The Design of Competitive Online Algorithms via a Primal–Dual Approach. *Foundations and Trends® in Theoretical Computer Science* 3, 2–3 (2009), 93–263.

[6] Eduardo F Camacho and Carlos Bordons Alba. 2013. *Model Predictive Control*. Springer Science & Business Media.

[7] Niangjun Chen, Anish Agarwal, Adam Wierman, Siddharth Barman, and Lachlan L.H. Andrew. 2015. Online Convex Optimization Using Predictions. *ACM SIGMETRICS Performance Evaluation Review* 43, 1 (2015), 191–204.

[8] Niangjun Chen, Joshua Comden, Zhenhua Liu, Anshul Gandhi, and Adam Wierman. 2016. Using Predictions in Online Optimization: Looking Forward with an Eye on the Past. *ACM SIGMETRICS Performance Evaluation Review* 44, 1 (2016), 193–206.

[9] Rami Cohen, Liane Lewin-Eytan, Joseph Seffi Naor, and Danny Raz. 2015. Near Optimal Placement of Virtual Network Functions. In *IEEE Conference on Computer Communications (INFOCOM)*. 1346–1354.

[10] Daniel Gmach, Yuan Chen, Amip Shah, Jerry Rolia, Cullen Bash, Tom Christian, and Ratnesh Sharma. 2010. Profiling Sustainability of Data Centers. In *IEEE International Symposium on Sustainable Systems and Technology (ISSST)*. 1–6.

[11] Mohammad H. Hajiesmaili, Chi-Kin Chau, Minghua Chen, and Longbu Huang. 2016. Online Microgrid Energy Generation Scheduling Revisited: The Benefits of Randomization and Interval Prediction. In *Proceedings of the 7th International Conference on Future Energy Systems (e-Energy '16)*. ACM, 1–11.

[12] Elad Hazan, Amit Agarwal, and Satyen Kale. 2007. Logarithmic Regret Algorithms for Online Convex Optimization. *Machine Learning* 69, 2-3 (2007), 169–192.

[13] T. S. Jayram, Tracy Kimbrel, Robert Krauthgamer, Baruch Schieber, and Maxim Sviridenko. 2001. Online Server Allocation in a Server Farm via Benefit Task Systems. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing (STOC '01)*. 540–549.

[14] Yongzheng Jia, Chuan Wu, Zongpeng Li, Franck Le, and Alex Liu. 2018. Online Scaling of NFV Service Chains Across Geo-Distributed Datacenters. *IEEE/ACM Transactions on Networking (TON)* 26, 2 (2018), 699–710.

[15] Lei Jiao, Lingjun Pu, Lin Wang, Xiaojun Lin, and Jun Li. 2018. Multiple Granularity Online Control of Cloudlet Networks for Edge Computing. In *15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*.

[16] Lei Jiao, Antonia Maria Tulino, Jaime Llorca, Yue Jin, and Alessandra Sala. 2017. Smoothed Online Resource Allocation in Multi-Tier Distributed Cloud Networks. *IEEE/ACM Transactions on Networking (TON)* 25, 4 (2017), 2556–2570.
[17] Lei Jiao, Ruiting Zhou, Xiaojun Lin, and Xu Chen. 2019. Online Scheduling of Traffic Diversion and Cloud Scrubbing with Uncertainty in Current Inputs. In *ACM MOBIHOC*.
[18] Vinay Joseph and Gustavo de Veciana. 2012. Jointly Optimizing Multi-User Rate Adaptation for Video Transport over Wireless Systems: Mean-Fairness-Variability Tradeoffs. In *IEEE Conference on Computer Communications (INFOCOM)*. 567–575.
[19] Seung-Jun Kim and Geogios B Giannakis. 2017. An Online Convex Optimization Approach to Real-Time Energy Pricing for Demand Response. *IEEE Transactions on Smart Grid* 8, 6 (2017), 2784–2793.
[20] Minghong Lin, Zhenhua Liu, Adam Wierman, and Lachlan LH Andrew. 2012. Online Algorithms for Geographical Load Balancing. In *International Green Computing Conference (IGCC)*. IEEE, 1–10.
[21] Minghong Lin, Adam Wierman, Lachlan L. H. Andrew, and Eno Thereska. 2013. Dynamic Right-sizing for Power-Proportional Data Centers. *IEEE/ACM Transactions on Networking (TON)* 21, 5 (2013), 1378–1391.
[22] Zhenhua Liu, Iris Liu, Steven Low, and Adam Wierman. 2014. Pricing Data Center Demand Response. *ACM SIGMETRICS Performance Evaluation Review* 42, 1 (2014), 111–123.
[23] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, and Niangjun Chen. 2013. Data Center Demand Response: Avoiding the Coincident Peak via Workload Shifting and Local Generation. *Performance Evaluation* 70, 10 (2013), 770–791.
[24] Lian Lu, Jinlong Tu, Chi-Kin Chau, Minghua Chen, and Xiaojun Lin. 2013. Online Energy Generation Scheduling for Microgrids with Intermittent Energy Sources and Co-Generation. *ACM SIGMETRICS Performance Evaluation Review* 41, 1 (2013), 53–66.
[25] Di Niu, Hong Xu, Baochun Li, and Shuqiao Zhao. 2012. Quality-Assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications.. In *IEEE Conference on Computer Communications (INFOCOM)*. 460–468.
[26] James Blake Rawlings and David Q Mayne. 2009. *Model Predictive Control: Theory and Design*. Nob Hill Pub. Madison, Wisconsin.
[27] Shai Shalev-Shwartz. 2012. Online Learning and Online Convex Optimization. *Foundations and Trends® in Machine Learning* 4, 2 (2012), 107–194.
[28] Ming Shi, Xiaojun Lin, Sonia Fahmy, and Dong-Hoon Shin. 2018. Competitive Online Convex Optimization with Switching Costs and Ramp Constraints. In *IEEE Conference on Computer Communications (INFOCOM)*. 1835–1843.
[29] Hao Wang, Jianwei Huang, Xiaojun Lin, and Hamed Mohsenian-Rad. 2014. Exploring Smart Grid and Data Center Interactions for Electric Power Load Balancing. *ACM SIGMETRICS Performance Evaluation Review* 41, 3 (2014), 89–94.
[30] Lin Xiao. 2010. Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization. *Journal of Machine Learning Research* 11, Oct (2010), 2543–2596.
[31] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. *ACM SIGCOMM Computer Communication Review* 45, 4 (2015), 325–338.
[32] Shizhen Zhao, Xiaojun Lin, Dionysios Aliprantis, Hugo N Villegas, and Minghua Chen. 2016. Online Multi-Stage Decisions for Robust Power-Grid Operations Under High Renewable Uncertainty. In *IEEE Conference on Computer Communications (INFOCOM)*. 1–9.
[33] Shizhen Zhao, Xiaojun Lin, and Minghua Chen. 2015. Peak-Minimizing Online EV Charging: Price-of-Uncertainty and Algorithm Robustification. In *IEEE Conference on Computer Communications (INFOCOM)*. 2335–2343.
[34] Martin Zinkevich. 2003. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML'03)*. AAAI Press, 928–935.

A DERIVATION OF THE DECISION SEQUENCE FOR AFHC IN COUNTER-EXAMPLE 1

In the following, we provide the derivation of the decision sequences of different versions of $\text{FHC}^{(\tau)}$.

$\text{FHC}^{(0)}$ sees the input sequence[2] $(0, \underline{a}, \underline{a}, \overline{a})$ in the first round[3] and gives the decision $(0, \underline{a}, \underline{a}, \overline{a})$ to satisfy the input $\overline{a}$ at time $t = 3$. It sees the input sequence $(\overline{a}, \underline{a}, \underline{a}, \overline{a})$ in all subsequent rounds and gives the decision sequence $(\overline{a}, \overline{a}, \overline{a}, \overline{a})$ for these rounds. To understand these decisions, note that the first and the fourth decisions should be $\overline{a}$ to satisfy the hard constraints. The second and third decisions are still $\overline{a}$ for the following reasons. First, $\text{FHC}^{(0)}$ knows that a large input $\overline{a}$ will be received at the end of the look-ahead window. Second, recall that the switching-cost coefficient w is much larger than the service-cost coefficient $c(t)$. Thus, if the second and third decisions were

[2]In the following, we divide the input and decision sequences into rounds of 4 time slots in order to describe their patterns.
[3]By convention, the input at time $t = 0$ is 0.

reduced, a large switching cost would be incurred later for increasing the decision back to $\overline{a}$. Hence, $(\overline{a}, \overline{a}, \overline{a}, \overline{a})$ are the optimal decisions for $\text{FHC}^{(0)}$ for these rounds.

$\text{FHC}^{(1)}$ sees the input sequence $(\underline{a}, \underline{a}, \overline{a}, \overline{a})$ and gives the decision sequence $(\underline{a}, \underline{a}, \overline{a}, \overline{a})$ in the first round. Note that the first two decisions are $\underline{a}$, which is enough to satisfy the inputs $\underline{a}$. The last two decisions need to increase to $\overline{a}$ to satisfy the inputs $\overline{a}$ at time $t = 3, 4$. $\text{FHC}^{(1)}$ sees the input sequence $(\underline{a}, \underline{a}, \overline{a}, \overline{a})$ in all subsequent rounds and gives the decision sequence $(\overline{a}, \overline{a}, \overline{a}, \overline{a})$ for these rounds. Similar to $\text{FHC}^{(0)}$ discussed above, since in each of these rounds, $\text{FHC}^{(1)}$ sees the decision $\overline{a}$ at the end of last round and knows that the large input $\overline{a}$ will be received at the end of the current round, $\text{FHC}^{(1)}$ will keep all the decisions to be $\overline{a}$.

Notice that both $\text{FHC}^{(0)}$ and $\text{FHC}^{(1)}$ follow exactly the same decisions as the optimal offline solution discussed in Sec. 3.1, because they see a large input $\overline{a}$ coming at the tail of the look-ahead window. In contrast, as we see below, $\text{FHC}^{(2)}$ and $\text{FHC}^{(3)}$ will have a different behavior due to the "end-of-look-ahead" problem that we mentioned before.

Specifically, $\text{FHC}^{(2)}$ sees the input sequence $(0, 0, 0, \underline{a})$ in the first round, and $(\underline{a}, \overline{a}, \overline{a}, \underline{a})$ in all subsequent rounds. Thus, $\text{FHC}^{(2)}$ gives the decision sequence $(0, 0, 0, \underline{a})$ in the first round and $(\underline{a}, \overline{a}, \overline{a}, \underline{a})$ in all subsequent rounds. To understand these decisions, for every round other than the first round, note that since the first input is $\underline{a}$ and the last time-slot's decision given by the previous round is $\underline{a}$, the first decision of each round is $\underline{a}$. Then, the decision needs to be increased to $\overline{a}$ to satisfy the second and third inputs. The last decision is $\underline{a}$ because future inputs are unknown to $\text{FHC}^{(2)}$, and thus using a lower decision value $\underline{a}$ does not incur any switching costs but lowers the service costs. (Recall that the switching costs are only incurred when the decision variables increase but not when they decrease.) As a result, $(\underline{a}, \overline{a}, \overline{a}, \underline{a})$ are the optimal decisions for $\text{FHC}^{(2)}$ at time $t = 4$, and get repeated every 4 time slots.

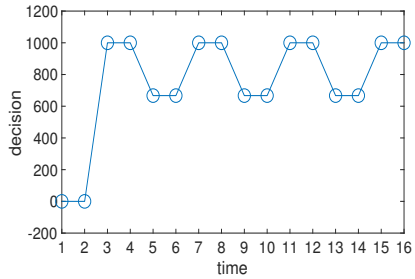
Similarly, $\text{FHC}^{(3)}$ sees the input sequence $(0, 0, \underline{a}, \underline{a})$ and gives the decision sequence $(0, 0, \underline{a}, \underline{a})$ in the first round. It sees input sequence $(\overline{a}, \overline{a}, \underline{a}, \underline{a})$ and gives the decision sequence $(\overline{a}, \overline{a}, \underline{a}, \underline{a})$ in all subsequent rounds. To understand these decisions, for every round other than the first round, note that the first two decisions need to be $\overline{a}$ to satisfy the first two inputs in each round. The last two decisions are $\underline{a}$ since it is sufficient to use a lower decision value in order to minimize the objective value.

All of these decision sequences are plotted in Fig.1d for $\underline{a} = 0$ and $\overline{a} = 1000$. We can notice that, while $\text{FHC}^{(0)}$ and $\text{FHC}^{(1)}$ keep the decision $\overline{a}$, the decisions of $\text{FHC}^{(2)}$ and $\text{FHC}^{(3)}$ go down in each round.

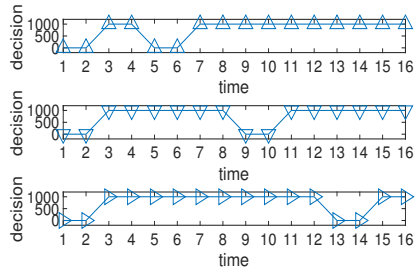
Finally, by taking the average of the decisions of these 4 versions of $\text{FHC}^{(\tau)}$, AFHC gives the final decision sequence $(\underline{a}, \underline{a}, \overline{a}, \overline{a})$ for the first round and $(\frac{\overline{a}+\underline{a}}{2}, \frac{\overline{a}+\underline{a}}{2}, \overline{a}, \overline{a})$ for all subsequent rounds (of every 4 slots) starting from time $t = 5$. The decision sequence is plotted in Fig. 1c for $\underline{a} = 0$ and $\overline{a} = 1000$.

B COUNTER-EXAMPLE 2

Note that the "Go-Back-1-Step" idea is somewhat similar to the idea of CHC (Committed Horizon Control) [8], where the decisions for the entire look-ahead window is computed, but only the first part of the decisions is committed. Although we could apply such an idea to AFHC directly, as we show below, it will still have the "end-of-look-ahead" problem as shown in Sec. 3.1. To see this, consider the following modification to AFHC and FHC, which we refer to as AFHC_{-1} and FHC_{-1}. Specifically, at each time $t = \tau + Ku$, $u = -1, 0, ..., \left\lfloor \frac{\mathcal{T}}{K} \right\rfloor$, $\text{FHC}_{-1}^{(\tau)}$ is given the current input and the inputs in the current look-ahead window $t + 1, ..., t + K$. $\text{FHC}_{-1}^{(\tau)}$ then calculates the solution to the optimization problem $Q^{\text{FHC}^{(\tau)}}(t)$ in Eq.(7), but it only commits to the first K decisions. Thus,



(a) AFHC_{-1}.



(b) Above: $\text{FHC}_{-1}^{(0)}$; Middle: $\text{FHC}_{-1}^{(1)}$; Below: $\text{FHC}_{-1}^{(2)}$.

Fig. 7. Behaviors of AFHC_{-1} with the same inputs as in Counter-example 1.

similar to RMHC, now the initial value $\vec{X}(t-1)$ for $Q^{\text{FHC}^{(\tau)}}(t)$ is from $\text{FHC}_{-1}^{(\tau)}$ computed at an earlier time of $t-K$, i.e., when $Q^{\text{FHC}^{(\tau)}}(t-K)$ was solved. AFHC_{-1} is formally given in Algorithm 4. We now use the same input sequence in Counter-example 1 to show that AFHC_{-1} can still have an arbitrarily large competitive ratio.

Algorithm 4 Averaging Fixed Horizon Control-first K decisions committed (AFHC_{-1})

Input: $\vec{A}(1:\mathcal{T})$, K
Output: $\vec{X}^{\text{AFHC}_{-1}}(1:\mathcal{T})$
FOR $t=-K+2:\mathcal{T}$
Step 1: Let $\tau \leftarrow t \bmod K$, and $\vec{X}(t-1) \leftarrow \vec{X}^{\text{FHC}_{-1}^{(\tau)}}(t-1)$.
Step 2: Based on $\vec{A}(t:t+K)$, solve problem $Q^{\text{FHC}^{(\tau)}}(t)$ in Eq.(7) and commits to the first K decisions, denoted as $\vec{X}^{\text{FHC}_{-1}^{(\tau)}}(t:t+K-1)$.
Step 3: Use Eq.(48) below to get the final decision

$$\vec{X}^{\text{AFHC}_{-1}}(t) = \frac{1}{K}\sum_{\tau=0}^{K-1} \vec{X}^{\text{FHC}_{-1}^{(\tau)}}(t). \tag{48}$$

END

Cost of AFHC_{-1}: Here, we consider the same input sequence as in Counter-example 1. Note that as $K=3$, there are 3 versions of FHC_{-1}. The decisions of $\text{FHC}_{-1}^{(0)}$, $\text{FHC}_{-1}^{(1)}$, $\text{FHC}_{-1}^{(2)}$ are plotted in Fig. 7b for $\underline{a}=0$ and $\overline{a}=1000$. Specifically, $\text{FHC}_{-1}^{(0)}$ sees input sequence $(\overline{a},\overline{a},\underline{a},\underline{a})$ in the round from time $t=3$ to $t=6$. Similar to $\text{FHC}^{(3)}$, it gives the decision sequence $(\overline{a},\overline{a},\underline{a},\underline{a})$ since there is no need to maintain a large decision at time $t=5,6$. However, based on the "Go-Back-1-Step" idea, it only commits to the first 3 slots, i.e., $(\overline{a},\overline{a},\underline{a})$. Then, at time $t=6$, it sees input $a(6)=\underline{a}$. Since $x(5)=\underline{a}$, it gives the decision $x(6)=\underline{a}$. In this way, a lower decision level $\underline{a}$ is still used at time $t=5$ and $t=6$. The same behavior occurs for $\text{FHC}_{-1}^{(1)}$ at time $t=9,10$ and for $\text{FHC}_{-1}^{(2)}$ at time $t=13,14$. When AFHC_{-1} takes the average of these 3 versions of $\text{FHC}_{-1}^{(\tau)}$, the final decision sequence becomes $(\underline{a},\underline{a},\overline{a},\overline{a})$ for the first round and $(\frac{2\overline{a}+\underline{a}}{3},\frac{2\overline{a}+\underline{a}}{3},\overline{a},\overline{a})$ for all subsequent rounds (of every 4 slots) starting from time $t=5$. We plot the final decision sequence of AFHC_{-1} in

Fig. 7a. The corresponding total cost is $\text{Cost}^{\text{AFHC}_{-1}} = [c(1)+c(2)]\,\underline{a} + [c(3)+c(4)]\,\overline{a} + w(\overline{a}-\underline{a}) + \sum_{j=2}^{\mathcal{T}/4}\left\{[c(4j-3)+c(4j-2)]\,\frac{2\overline{a}+\underline{a}}{3} + \big[c(4j-1)+c(4j)\big]\overline{a} + w\frac{\overline{a}-\underline{a}}{3}\right\}$. Notice that the total switching cost of AFHC_{-1} still increases by a value of $w\frac{\overline{a}-\underline{a}}{3}$ every 4 time slots due to the terms $\sum_{j=2}^{\mathcal{T}/4} w\frac{\overline{a}-\underline{a}}{3}$. Thus, similar to Counter-example 1, if $w \geq \sum_{t=1}^{\mathcal{T}} c(t)$, the ratio between the total cost of AFHC_{-1} and that of the optimal offline solution is

$$\frac{\text{Cost}^{\text{AFHC}_{-1}}(1:\mathcal{T})}{\text{Cost}^{\text{OPT}}(1:\mathcal{T})} \geq \frac{\frac{\mathcal{T}}{4}w\frac{\overline{a}-\underline{a}}{3}}{2w\overline{a}} = \frac{\mathcal{T}}{24}\frac{\overline{a}-\underline{a}}{\overline{a}}, \tag{49}$$

which can become arbitrarily large as the time horizon $\mathcal{T}$ increases.

C PROOF OF LEMMA 3.2

Proof. Notice that each version of $\text{RMHC}^{(\tau)}$ solves the optimization problem (9) to get the corresponding decisions $\vec{X}^{\text{RMHC}^{(\tau)}}(t:t+K-1)$ and $\vec{Y}^{\text{RMHC}^{(\tau)}}(t+K-1)$. This optimization problem (9) is equivalent to the optimization problem (16). Then, by applying the Karush-Kuhn-Tucker (KKT) conditions [2, p. 243] to (16), we have the following equations:

Complementary slackness:

$$\big(\vec{\beta}^{\text{RMHC}^{(\tau)}}(s)\big)^{\text{T}}[\boldsymbol{B_2}\vec{A}(s) - \boldsymbol{B_1}\vec{X}^{\text{RMHC}^{(\tau)}}(s)] = 0, \text{ for all time } s \in [t, t+K-1], \tag{50a}$$

$$\big(\vec{\theta}^{\text{RMHC}^{(\tau)}}(s)\big)^{\text{T}}[\vec{X}^{\text{RMHC}^{(\tau)}}(s) - \vec{X}^{\text{RMHC}^{(\tau)}}(s-1) - \vec{Y}^{\text{RMHC}^{(\tau)}}(s)] = 0, \quad \text{for all time } s \in [t, t+K-1], \tag{50b}$$

Stationarity/Optimality:

$$\big(\vec{X}^{\text{RMHC}^{(\tau)}}(s)\big)^{\text{T}}[\vec{C}(s) - \boldsymbol{B_1}^{\text{T}}\vec{\beta}^{\text{RMHC}^{(\tau)}}(s) + \vec{\theta}^{\text{RMHC}^{(\tau)}}(s) - \vec{\theta}^{\text{RMHC}^{(\tau)}}(s+1)] = 0, \quad \text{for all time } s \in [t, t+K-2], \tag{51a}$$

$$\big(\vec{X}^{\text{RMHC}^{(\tau)}}(t+K-1)\big)^{\text{T}}\big[\vec{C}(t+K-1) - \boldsymbol{B_1}^{\text{T}}\vec{\beta}^{\text{RMHC}^{(\tau)}}(t+K-1) + \vec{\theta}^{\text{RMHC}^{(\tau)}}(t+K-1) - [\vec{W} - \vec{C}(t+K)]^{+}\big] = 0, \tag{51b}$$

$$\big(\vec{Y}^{\text{RMHC}^{(\tau)}}(s)\big)^{\text{T}}[\vec{W} - \vec{\theta}^{\text{RMHC}^{(\tau)}}(s)] = 0, \text{ for all time } s \in [t, t+K-1]. \tag{51c}$$

We now calculate the real cost of each version of $\text{RMHC}^{(\tau)}$ in the current decision round from time t to $t+K-1$, where $t = \tau + Ku$ and $u = -1, 0, \ldots, \lfloor \frac{\mathcal{T}}{K} \rfloor$. We have

$$\begin{aligned}
&\text{Cost}^{\text{RMHC}^{(\tau)}}(t:t+K-1) \\
&= \sum_{s=t}^{t+K-1}\left\{\vec{C}^{\text{T}}(s)\vec{X}^{\text{RMHC}^{(\tau)}}(s) + \vec{W}^{\text{T}}[\vec{X}^{\text{RMHC}^{(\tau)}}(s) - \vec{X}^{\text{RMHC}^{(\tau)}}(s-1)]^{+}\right\} \\
&\leq \sum_{s=t}^{t+K-1}\left\{\vec{C}^{\text{T}}(s)\vec{X}^{\text{RMHC}^{(\tau)}}(s) + \vec{W}^{\text{T}}\vec{Y}^{\text{RMHC}^{(\tau)}}(s)\right\},
\end{aligned} \tag{52}$$

where the last inequality is due to constraints (16c) and (16d). Then, according to Eqs.(50a) and (50b), by adding these zero-value terms, adding and deducting the term $[\vec{W} - \vec{C}(t+K)]^{+}\vec{X}^{\text{RMHC}^{(\tau)}}(t+K-1)$,

we have

$$\begin{aligned}
&\mathrm{Cost}^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1)\\
&\le \sum_{s=t}^{t+K-1}\left\{\vec{C}^{\mathrm{T}}(s)\vec{X}^{\mathrm{RMHC}^{(\tau)}}(s)+\vec{W}^{\mathrm{T}}\vec{Y}^{\mathrm{RMHC}^{(\tau)}}(s)\right\}\\
&\quad+\sum_{s=t}^{t+K-1}\left\{\left(\vec{\beta}^{\mathrm{RMHC}^{(\tau)}}(s)\right)^{\mathrm{T}}\left[\boldsymbol{B_2}\vec{A}(s)-\boldsymbol{B_1}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(s)\right]\right\}\\
&\quad+\sum_{s=t}^{t+K-1}\left\{\left(\vec{\theta}^{\mathrm{RMHC}^{(\tau)}}(s)\right)^{\mathrm{T}}[\vec{X}^{\mathrm{RMHC}^{(\tau)}}(s)-\vec{X}^{\mathrm{RMHC}^{(\tau)}}(s-1)-\vec{Y}^{\mathrm{RMHC}^{(\tau)}}(s)]\right\}\\
&\quad+[\vec{W}-\vec{C}(t+K)]^{+}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t+K-1)-[\vec{W}-\vec{C}(t+K)]^{+}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t+K-1).
\end{aligned}\tag{53}$$

Rearranging Eq.(53), we get

$$\begin{aligned}
&\mathrm{Cost}^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1)\\
&\le \sum_{s=t}^{t+K-1}\left(\vec{\beta}^{\mathrm{RMHC}^{(\tau)}}(s)\right)^{\mathrm{T}}\boldsymbol{B_2}\vec{A}(s)+[\vec{W}^{\mathrm{T}}-\vec{C}^{\mathrm{T}}(t+K)]^{+}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t+K-1)\\
&\quad+\sum_{s=t}^{t+K-2}\left(\vec{X}^{\mathrm{RMHC}^{(\tau)}}(s)\right)^{\mathrm{T}}\left[\vec{C}(s)-\boldsymbol{B_1}^{\mathrm{T}}\vec{\beta}^{\mathrm{RMHC}^{(\tau)}}(s)+\vec{\theta}^{\mathrm{RMHC}^{(\tau)}}(s)-\vec{\theta}^{\mathrm{RMHC}^{(\tau)}}(s+1)\right]\\
&\quad+\left(\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t+K-1)\right)^{\mathrm{T}}\Big[\vec{C}(t+K-1)\\
&\qquad\qquad-\boldsymbol{B_1}^{\mathrm{T}}\vec{\beta}^{\mathrm{RMHC}^{(\tau)}}(t+K-1)+\vec{\theta}^{\mathrm{RMHC}^{(\tau)}}(t+K-1)-[\vec{W}-\vec{C}(t+K)]^{+}\Big]\\
&\quad+\sum_{s=t}^{t+K-1}\left(\vec{Y}^{\mathrm{RMHC}^{(\tau)}}(s)\right)^{\mathrm{T}}\left[\vec{W}-\vec{\theta}^{\mathrm{RMHC}^{(\tau)}}(s)\right]-\vec{W}^{\mathrm{T}}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t-1).
\end{aligned}$$

Using Eqs.(51a-51c), all except the first two terms and the last term on the right-hand side are equal to 0. Finally, using the notations in Eqs.(18) and (19a), (19b), we have

$$\begin{aligned}
&\mathrm{Cost}^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1)\\
&\le D^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1)-\phi^{(\tau)}(t-1)+\psi^{(\tau)}(t+K-1).
\end{aligned}\tag{55}$$

□

D PROOF OF LEMMA 3.3

Proof. **Part (1) of Lemma 3.3:** first, since there are K versions of $\mathrm{RMHC}^{(\tau)}$, there are K rows in the original matrix G^{old}. The total time horizon is at most from $t=-K+2$ to $t=\mathcal{T}$ and $\mathrm{RMHC}^{(\tau)}$ proceeds every K time slots. Thus, there are at most $\left\lfloor\frac{\mathcal{T}}{K}\right\rfloor+2$ components in each row of G^{old}. Hence, the sum on the left-hand side of Eq.(24) represents the sum of all the components in the original matrix G^{old}.

Second, recall that some components from G^{old} are skipped during re-stitching and are concatenated in the $(K+1)$-th row of G^{new}. Thus, there are $K+1$ rows in the new matrix G^{new}. Between every two adjacent components, we add a zero-value component corresponding to the new dual variables added during re-stitching. Thus, there are at most $2\left(\left\lfloor\frac{\mathcal{T}}{K+1}\right\rfloor+2\right)-1=2\left\lfloor\frac{\mathcal{T}}{K+1}\right\rfloor+3$ components in each row of the new matrix G^{new}. Hence, the sum on the right-hand side of Eq.(24) represents the sum of all the components in the new group G^{new}.

Finally, during re-stitching, we only add non-negative components and do not remove any components or terms in the original group G^{old}. Hence, the sum of the components in corresponding groups will not decrease. Eq.(24) is thus proven.

Part (2) of Lemma 3.3: notice that in each row of G^{new}, K online dual costs $D^{\text{RMHC}^{(\tau)}}(t)$ and a zero term appear repeatedly. Without loss of generality, consider the online dual costs $D^{\text{RMHC}^{(\tau)}}(t : t+K-1)$, followed by a zero, and then followed by $D^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1 : t+2K)$, where $t = \tau + Ku$ and $u = -1, 0, ..., \lfloor \frac{\mathcal{T}}{K} \rfloor$. We only need to prove that the dual variables from time t to time $t+K$ are feasible to the offline dual optimization problem (14). The feasibility will then hold for all other time slots simply by repeating the argument in every round of $K+1$ time slots (e.g., from time $t-K-1$ to $t-1$ or from time $t+K+1$ to $t+2K+1$).

Towards this end, we first provide some properties of the online dual variables $\vec{\beta}^{\text{RMHC}^{(\tau)}}(t : t+K-1)$ and $\vec{\theta}^{\text{RMHC}^{(\tau)}}(t : t+K-1)$. Recall that the optimization problem (16) is a minimization problem. Then, by the definition of the dual, we have

$$\vec{\beta}^{\text{RMHC}^{(\tau)}}(s) \geq 0, \text{ for all time } s \in [t, t+K-1], \tag{56a}$$

$$\vec{\theta}^{\text{RMHC}^{(\tau)}}(s) \geq 0, \text{ for all time } s \in [t, t+K-1]. \tag{56b}$$

Similarly, from the stationarity/optimality conditions (51a-51c) in Appendix C, we know that these online dual variables from each version of $\text{RMHC}^{(\tau)}$ satisfy the following dual constraints,

$$\vec{C}(s) - \boldsymbol{B}_1^{\text{T}} \vec{\beta}^{\text{RMHC}^{(\tau)}}(s) + \vec{\theta}^{\text{RMHC}^{(\tau)}}(s) - \vec{\theta}^{\text{RMHC}^{(\tau)}}(s+1) \geq 0, \text{ for all time } s \in [t, t+K-2], \tag{57a}$$

$$\vec{C}(t+K-1) - \boldsymbol{B}_1^{\text{T}} \vec{\beta}^{\text{RMHC}^{(\tau)}}(t+K-1) + \vec{\theta}^{\text{RMHC}^{(\tau)}}(t+K-1) - [\vec{W} - \vec{C}(t+K)]^+ \geq 0, \tag{57b}$$

$$\vec{W} - \vec{\theta}^{\text{RMHC}^{(\tau)}}(s) \geq 0, \quad \text{for all time } s \in [t, t+K-1]. \tag{57c}$$

Then, we can verify the feasibility of the online dual variables from time t to time $t+K$ for the offline dual optimization problem (14) as follows.

(i) Consider time t to time $t+K-2$. The online dual variables $\vec{\beta}^{\text{RMHC}^{(\tau)}}(t : t+K-2)$ and $\vec{\theta}^{\text{RMHC}^{(\tau)}}(t : t+K-2)$ satisfy constraints (56a) and (56b), Eqs.(57a) and (57c). Thus, they are feasible to constraints (14b-14d) from time t to $t+K-2$ in the offline dual optimization problem (14).

(ii) For time $t+K-1$, constraints (14b) and (14d) still hold due to Eqs.(56a), (56b) and (57c). Then, consider constraint (14c) with the online dual variables $\vec{\beta}^{\text{RMHC}^{(\tau)}}(t+K-1)$, $\vec{\theta}^{\text{RMHC}^{(\tau)}}(t+K-1)$ and $\tilde{\vec{\theta}}(t+K)$. The left-hand side of constraint (14c) satisfies

$$\vec{C}(t+K-1) - \boldsymbol{B}_1^{\text{T}} \vec{\beta}^{\text{RMHC}^{(\tau)}}(t+K-1) + \vec{\theta}^{\text{RMHC}^{(\tau)}}(t+K-1) - \tilde{\vec{\theta}}(t+K) \tag{58a}$$

$$= \vec{C}(t+K-1) - \boldsymbol{B}_1^{\text{T}} \vec{\beta}^{\text{RMHC}^{(\tau)}}(t+K-1) + \vec{\theta}^{\text{RMHC}^{(\tau)}}(t+K-1) - [\vec{W} - \vec{C}(t+K)]^+ \tag{58b}$$

$$\geq 0. \tag{58c}$$

Eq.(58b) holds because we add $\tilde{\vec{\theta}}(t+K) = [\vec{W} - \vec{C}(t+K)]^+$ during the re-stitching step. Eq.(58c) is true due to constraint (57b).

(iii) Similarly, for time $t+K$, constraints (14b) and (14d) still hold. Further, the left-hand side of constraint (14c) satisfies

$$\begin{aligned} &\vec{C}(t+K) - \boldsymbol{B}_1^{\text{T}} \tilde{\vec{\beta}}(t+K) + \tilde{\vec{\theta}}(t+K) - \vec{\theta}^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1) \\ &= \vec{C}(t+K) - 0 + [\vec{W} - \vec{C}(t+K)]^+ - \vec{\theta}^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1) \end{aligned} \tag{59}$$

where Eq.(59) is due to $\tilde{\vec{\beta}}(t+K) = 0, \tilde{\vec{\theta}}(t+K) = [\vec{W} - \vec{C}(t+K)]^+$. We now show that $\vec{\theta}^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1) = \vec{W}$. Suppose in the contrary that $\vec{\theta}^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1) < \vec{W}$. First, notice that if we increase $\vec{\theta}^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1)$ by a value of δ, we can also increase $\vec{\beta}^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1)$ such that $\boldsymbol{B}_1^{\text{T}}\vec{\beta}^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1)$ increases by the same value of δ. Then, constraint (14c) still holds. However, by doing so, we get a larger value of the dual objective $\sum_{t=1}^{\mathcal{T}} \vec{\beta}^{\text{T}}(t)\boldsymbol{B}_2\vec{A}(t)$. This contradicts the fact that $\vec{\theta}^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1)$ is the optimal dual solution to RMHC. Thus, we must have $\vec{\theta}^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1) = \vec{W}$. From Eq.(59), we thus have,

$$\begin{aligned}
&\vec{C}(t+K) - \boldsymbol{B}_1^{\text{T}}\tilde{\vec{\beta}}(t+K) + \tilde{\vec{\theta}}(t+K) - \vec{\theta}^{\text{RMHC}^{((\tau+1) \bmod K)}}(t+K+1) \\
&\geq \vec{C}(t+K) - 0 + \vec{W} - \vec{C}(t+K) - \vec{W} \\
&= 0,
\end{aligned}$$

Hence, the online dual variables from each row of the new matrix G^{new} are feasible to the offline dual optimization problem (14).

□

E PROOF OF LEMMA 3.4

Proof. Applying the definition of $\phi^{(\tau)}(t-1)$ and $\psi^{(\tau)}(t+K-1)$ in Eq.(19), we have that, for any $\tau \in [0, K-1]$, the sum of the tail-terms for all rounds satisfies

$$\sum_{u=-1}^{\lfloor \frac{\mathcal{T}}{K} \rfloor} \left\{ \phi^{(\tau)}(\tau + Ku - 1) + \psi^{(\tau)}(\tau + Ku + K - 1) \right\} \tag{61a}$$

$$= \sum_{u=-1}^{\lfloor \frac{\mathcal{T}}{K} \rfloor} \left\{ -\vec{W}^{\text{T}}\vec{X}^{\text{RMHC}^{(\tau)}}(\tau + Ku - 1) + [\vec{W}^{\text{T}} - \vec{C}^{\text{T}}(\tau + Ku + K)]^+ \cdot \vec{X}^{\text{RMHC}^{(\tau)}}(\tau + Ku + K - 1) \right\}. \tag{61b}$$

We call the two tail-terms in Eq.(61b) with the same index u as a "pair". Then, we rearrange the terms in the summation by putting together the second term in the previous pair and the first term in the next pair. Notice that, in the first pair, the first term $\phi^{(\tau)}(\tau - K - 1) = -\vec{W}^{\text{T}}\vec{X}^{\text{RMHC}^{(\tau)}}(\tau - K - 1) = 0$. This is because, by convention, $\vec{X}(t) = 0$ for all time $t \leq 0$. In the last pair, the second term $\psi^{(\tau)}(\tau + K\lfloor \frac{\mathcal{T}}{K} \rfloor + K - 1) = [\vec{W}^{\text{T}} - \vec{C}^{\text{T}}(\tau + K\lfloor \frac{\mathcal{T}}{K} \rfloor + K)]^+\vec{X}^{\text{RMHC}^{(\tau)}}(\tau + K\lfloor \frac{\mathcal{T}}{K} \rfloor + K - 1) = 0$. This is because, when $t + K > T$, RMHC solves problem $\tilde{Q}^{\text{RMHC}^{(\tau)}}(t)$ in Eq.(10) instead of $Q^{\text{RMHC}^{(\tau)}}(t)$ in Eq.(9), which takes the future inputs after time $\mathcal{T}$ as 0. Thus, we have

$$\sum_{u=-1}^{\lfloor \frac{\mathcal{T}}{K} \rfloor} \left\{ \phi^{(\tau)}(\tau + Ku - 1) + \psi^{(\tau)}(\tau + Ku + K - 1) \right\}$$

$$= \sum_{u=0}^{\lfloor \frac{\mathcal{T}}{K} \rfloor - 1} \left\{ -\vec{W}^{\text{T}}\vec{X}^{\text{RMHC}^{(\tau)}}(\tau + Ku - 1) + [\vec{W}^{\text{T}} - \vec{C}^{\text{T}}(\tau + Ku)]^+ \cdot \vec{X}^{\text{RMHC}^{(\tau)}}(\tau + Ku - 1) \right\} \tag{62a}$$

$$\leq 0, \tag{62b}$$

where Eq.(62b) is due to

$$\begin{aligned}
&[\vec{W}^{\text{T}} - \vec{C}^{\text{T}}(\tau + Ku)]^+\vec{X}^{\text{RMHC}^{(\tau)}}(\tau + Ku - 1) \\
&\leq \vec{W}^{\text{T}}\vec{X}^{\text{RMHC}^{(\tau)}}(\tau + Ku - 1).
\end{aligned} \tag{63}$$

The result of Lemma 3.4 then follows. □

F PROOF OF THEOREM 3.5

The proof follows a similar line as in steps 1, 2 and 3 in the proof of Theorem 3.1 in Sec. 3.3. We first provide some preliminary results.

F.1 Preliminary Results

Notice that in Sec. 3.3, by introducing an auxiliary variable $\vec{Y}(t)$ for the switching term $[\vec{X}(t) - \vec{X}(t-1)]^+$, we can get an equivalent formulation (13) for the original offline optimization problem (2). Similarly, we can apply this method to get the following equivalent formulation for the offline optimization problem with convex service costs,

$$\min_{\{\vec{X}(1:\mathcal{T}),\vec{Y}(1:\mathcal{T})\}} \left\{ \sum_{t=1}^{\mathcal{T}} h_t(\vec{X}(t)) + \sum_{t=1}^{\mathcal{T}} \vec{W}^{\mathrm{T}}\vec{Y}(t) \right\} \tag{64a}$$

$$\text{sub. to: } \boldsymbol{B}_1\vec{X}(t) \geq \boldsymbol{B}_2\vec{A}(t), \text{ for all time } t \in [1,\mathcal{T}], \tag{64b}$$

$$\vec{Y}(t) \geq \vec{X}(t) - \vec{X}(t-1), \text{ for all time } t \in [1,\mathcal{T}], \tag{64c}$$

$$\vec{X}(t), \vec{Y}(t) \geq 0, \text{ for all time } t \in [1,\mathcal{T}]. \tag{64d}$$

Then, let $\vec{\beta}(t) = [\beta_l(t), l = 1,...,L]^{\mathrm{T}} \in \mathbb{R}^{L\times 1}$ and $\vec{\theta}(t) = [\theta_n(t), n = 1,...,N]^{\mathrm{T}} \in \mathbb{R}^{N\times 1}$ be the Lagrange multipliers for constraints (64b) and (64c), respectively. We have the Lagrangian of problem (64) as follows,

$$\begin{aligned} &L(\vec{X}(1:\mathcal{T}),\vec{Y}(1:\mathcal{T}),\vec{\beta}(1:\mathcal{T}),\vec{\theta}(1:\mathcal{T})) \\ &= \sum_{t=1}^{T} h_t(\vec{X}(t)) + \sum_{t=1}^{T} \vec{W}^{\mathrm{T}}\vec{Y}(t) + \sum_{t=1}^{T} \vec{\beta}^{\mathrm{T}}(t)\left[\boldsymbol{B}_2\vec{A}(t) - \boldsymbol{B}_1\vec{X}(t)\right] + \sum_{t=1}^{T} \vec{\theta}^{\mathrm{T}}(t)\left[\vec{X}(t) - \vec{X}(t-1) - \vec{Y}(t)\right]. \end{aligned} \tag{65}$$

Step-1: Similar to Eq.(64), we can get the following equivalent formulation for the optimization problem (39) with convex service costs,

$$\min_{\substack{\{\vec{X}(t:t+K-1),\\ \vec{Y}(t:t+K-1)\}}} \left\{ \sum_{s=t}^{t+K-1} \left[h_s(\vec{X}(s)) + \vec{W}^{\mathrm{T}}\vec{Y}(s)\right] - \vec{W}^{\mathrm{T}}\vec{X}(t+K-1) \right\} \tag{66a}$$

$$\text{sub. to: } \boldsymbol{B}_1\vec{X}(s) \geq \boldsymbol{B}_2\vec{A}(s), \text{ for all time } s \in [t, t+K-1], \tag{66b}$$

$$\vec{Y}(s) \geq \vec{X}(s) - \vec{X}(s-1), \text{ for all time } s \in [t, t+K-1], \tag{66c}$$

$$\vec{X}(s), \vec{Y}(s) \geq 0, \text{ for all time } s \in [t, t+K-1]. \tag{66d}$$

Now, for each round of RMHC$^{(\tau)}$ from time t to $t+K-1$, where $t = \tau + Ku$ and $u = -1, 0, ..., \lfloor \frac{\mathcal{T}}{K} \rfloor$, the new formulation of the online primal cost is

$$\mathrm{Cost}^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1) = \sum_{s=t}^{t+K-1} \left\{ h_s(\vec{X}^{\mathrm{RMHC}^{(\tau)}}(s)) + \vec{W}^{\mathrm{T}}[\vec{X}^{\mathrm{RMHC}^{(\tau)}}(s) - \vec{X}^{\mathrm{RMHC}^{(\tau)}}(s)]^+ \right\}. \tag{67}$$

Next, let $\left(\vec{\beta}^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1), \vec{\theta}^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1)\right)$ be the optimal dual solution to the dual of the optimization problem in Eq.(66). Then, for each $\tau \in [0, K-1]$, we can define the online dual

cost by

$$\begin{aligned} & D^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1) \\ &= \sum_{s=t}^{t+K-1} \left(\vec{\beta}^{\mathrm{RMHC}^{(\tau)}}(s)\right)^{\mathrm{T}} \boldsymbol{B_2}\vec{A}(s) \\ &\quad + \sum_{s=t}^{t+K-2} \eta_s \left(-\boldsymbol{B_1}^{\mathrm{T}}\vec{\beta}^{\mathrm{RMHC}^{(\tau)}}(s) + \vec{\theta}^{\mathrm{RMHC}^{(\tau)}}(s) - \vec{\theta}^{\mathrm{RMHC}^{(\tau)}}(s+1)\right) \\ &\quad + \eta_{t+K-1}\left(-\boldsymbol{B_1}^{\mathrm{T}}\vec{\beta}^{\mathrm{RMHC}^{(\tau)}}(t+K-1) + \vec{\theta}^{\mathrm{RMHC}^{(\tau)}}(t+K-1) - \vec{W}\right), \end{aligned} \tag{68}$$

where

$$\eta_s(\vec{u}) = \min_{\vec{X}\geq 0}\left\{h_s(\vec{X}) + \vec{u}^{\mathrm{T}}\vec{X}\right\}. \tag{69}$$

Further, we can define the following tail-terms as in (19),

$$\phi^{(\tau)}(t-1) = -\vec{W}^{\mathrm{T}}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t-1), \tag{70a}$$

$$\psi^{(\tau)}(t+K-1) = \vec{W}^{\mathrm{T}}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t+K-1). \tag{70b}$$

Then, we have Lemma F.1 below, which is similar to Lemma 3.2.

Lemma F.1. *(Per-round gap between the primal and dual costs of RMHC$^{(\tau)}$) Using the new definition of Cost$^{RMHC^{(\tau)}}(t:t+K-1)$, $D^{RMHC^{(\tau)}}(t:t+K-1)$ and the tail-terms in Eqs.(67-70), for each $\tau\in[0,K-1]$ and for any $t=\tau+Ku$ where $u=-1,0,\ldots,\left\lfloor\frac{\mathcal{T}}{K}\right\rfloor$, we have*

$$\begin{aligned} Cost^{RMHC^{(\tau)}}(t:t+K-1) &\leq D^{RMHC^{(\tau)}}(t:t+K-1) \\ &+ \phi^{(\tau)}(t-1) + \psi^{(\tau)}(t+K-1), \text{ for all } \vec{A}(t:t+K-1). \end{aligned} \tag{71}$$

The proof uses duality theorem [2, p. 260] and follows the same line as in the proof of Lemma 3.2 in Appendix C.

Step-2: From Eqs.(64) and (65), we know that the optimal offline dual cost is

$$D^{\mathrm{OPT}}(1:\mathcal{T}) = \max_{\substack{\{\vec{\beta}(1:\mathcal{T})\geq 0,\\ \vec{\theta}(1:\mathcal{T})\geq 0\}}} \min_{\substack{\{\vec{X}(1:\mathcal{T}),\\ \vec{Y}(1:\mathcal{T})\}}} L(\vec{X}(1:\mathcal{T}),\vec{Y}(1:\mathcal{T}),\vec{\beta}(1:\mathcal{T}),\vec{\theta}(1:\mathcal{T})). \tag{72}$$

We wish to construct online dual variables that are feasible for the offline optimization problem (64). Towards this end, we first characterize a condition for the dual feasibility of the offline problem. Recall that, given the dual variables $\vec{\beta}(1:\mathcal{T})$ and $\vec{\theta}(1:\mathcal{T})$, the dual objective function of the offline problem is given by

$$\min_{\{\vec{X}(1:\mathcal{T}),\vec{Y}(1:\mathcal{T})\}} L(\vec{X}(1:\mathcal{T}),\vec{Y}(1:\mathcal{T}),\vec{\beta}(1:\mathcal{T}),\vec{\theta}(1:\mathcal{T})) \tag{73}$$

We say that the dual variables $\vec{\beta}(1:\mathcal{T})$ and $\vec{\theta}(1:\mathcal{T})$ are feasible if the above dual objective function is not $-\infty$.

Collecting all the terms of $L(\vec{X}(1:\mathcal{T}),\vec{Y}(1:\mathcal{T}),\vec{\beta}(1:\mathcal{T}),\vec{\theta}(1:\mathcal{T}))$ that depend on $\vec{X}(t)$ and $\vec{Y}(t)$, and checking when the corresponding minimization will approach $-\infty$. we can show that the dual variables $(\vec{\beta}(1:\mathcal{T}),\vec{\theta}(1:\mathcal{T}))$ are feasible for the offline problem if they satisfy the following three conditions simultaneously for all $t\in[1,\mathcal{T}]$,

$$\vec{\beta}(t)\geq 0, \vec{\theta}(t)\geq 0, \tag{74a}$$

$$\min_{\vec{X}(t)\geq 0}\left\{h_t(\vec{X}(t))+\left[-\vec{\beta}^{\mathrm{T}}(t)\boldsymbol{B_1}+\vec{\theta}^{\mathrm{T}}(t)-\vec{\theta}^{\mathrm{T}}(t+1)\right]\cdot\vec{X}(t)\right\}>-\infty, \tag{74b}$$

$$\vec{W}-\vec{\theta}(t)\geq 0. \tag{74c}$$

We now construct online dual variables that satisfy (74). First, from the construction of the online dual $D^{\mathrm{RMHC}^{(\tau)}}(t:t+K-1)$ in Eq.(68), we can easily verify that, because the online dual variables chosen by $\mathrm{RMHC}^{(\tau)}$ should make (68) not equal to $-\infty$, they must also satisfy constraints (74a-74c) for time t to $t+K-2$. Second, applying the re-stitching idea, we can organize these online duals into the matrix G^{new}. Then, in the i-th row of G^{new}, we introduce extra dual variables (corresponds to the 0 terms in G^{new})

$$\begin{cases}\tilde{\vec{\beta}}(i+(K+1)u+K)=\vec{0},\\ \tilde{\vec{\theta}}(i+(K+1)u+K)=\vec{W}\end{cases} \tag{75}$$

at time $i+(K+1)u+K$ for every $u=-1,0,\ldots,\left\lfloor\frac{\mathcal{T}}{K}\right\rfloor$. Then, we can verify Lemma 3.3 for time $t+K-1$ and time $t+K$ as follows. Let $t=i+(K+1)u$ for some u and consider the round of $\mathrm{RMHC}^{(\tau)}$ starting from t. First, at time $s=t+K-1$, constraints (74a) and (74c) trivially hold because the corresponding dual variables at $t+K-1$ satisfies the KKT conditions for $\mathrm{RMHC}(i)$ (see Eqs.(56a), (56b) and (57c)). Further, for this time slot, the corresponding term in (68) is identical to the corresponding term in the offline dual. Thus, because our online dual variables makes (68) not equal to $-\infty$, they must satisfying constraint (74b). Second, at time $t+K$, the corresponding term in the offline Lagrangian is

$$\begin{aligned}&h_{t+K}(\vec{X}(t+K))+\tilde{\vec{\beta}}^{\mathrm{T}}(t+K)\boldsymbol{B_2}\vec{A}(t+K)\\&\quad+\left[-\boldsymbol{B_1}^{\mathrm{T}}\tilde{\vec{\beta}}(t+K)+\tilde{\vec{\theta}}(t+K)-\vec{\theta}(t+K+1)\right]^{\mathrm{T}}\vec{X}(t+K).\end{aligned} \tag{76}$$

Note that we have set $\vec{\beta}(t+K)=\vec{0}$ and $\vec{\theta}(t+K)=\vec{W}$. Further, we can show that $\vec{\theta}(t+K+1)=\vec{W}$ (see Appendix D). Thus, this term in the Lagrangian corresponding to time $t+K$ is simply $h_{t+K}(\vec{X}(t+K))$. Since $h_{t+K}(\vec{X}(t+K))$ is assumed to be non-negative, the constraint (74b) is trivially satisfied. Part (2) of Lemma 3.3 thus holds. Finally, the corresponding offline dual cost for time $t+K$ must also be non-negative. Thus, following the proof in Appendix D, we still have the conclusion in Lemma 3.3.

Step-3: Notice that, by replacing $[\vec{W}^{\mathrm{T}}-\vec{C}^{\mathrm{T}}(\tau+Ku)]^{+}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(\tau+Ku-1)$ with $\vec{W}^{\mathrm{T}}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(\tau+Ku-1)$ in Appendix E, each step in the proof in Appendix E still holds. Hence, Lemma 3.4 still holds.

F.2 Proof of Theorem 3.5

Proof. First, since now the service cost is convex, the total cost of ARMHC is

$$\mathrm{Cost}^{\mathrm{ARMHC}}(1:\mathcal{T})=\sum_{t=1}^{\mathcal{T}}h_t(\vec{X}^{\mathrm{ARMHC}}(t))+\sum_{t=1}^{\mathcal{T}}\vec{W}^{\mathrm{T}}[\vec{X}^{\mathrm{ARMHC}}(t)-\vec{X}^{\mathrm{ARMHC}}(t-1)]^{+}.$$

Then, from Eq.(11) that ARMHC takes the average of K versions of RMHC decisions, we have

$$\begin{aligned}\mathrm{Cost}^{\mathrm{ARMHC}}(1:\mathcal{T})&=\sum_{t=1}^{\mathcal{T}}h_t\left(\frac{1}{K}\sum_{\tau=0}^{K-1}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t)\right)\\&\quad+\sum_{t=1}^{\mathcal{T}}\vec{W}^{\mathrm{T}}\left[\frac{1}{K}\sum_{\tau=0}^{K-1}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t)-\frac{1}{K}\sum_{\tau=0}^{K-1}\vec{X}^{\mathrm{RMHC}^{(\tau)}}(t-1)\right]^{+}.\end{aligned}$$

Note that both $h_t(\cdot)$ and $[\cdot]^+$ are convex. According to Jensen's Inequality, we have

$$\begin{aligned}\text{Cost}^{\text{ARMHC}}(1:\mathcal{T}) \leq \frac{1}{K}\sum_{t=1}^{T}\sum_{\tau=0}^{K-1} h_t(\vec{X}^{\text{RMHC}^{(\tau)}}(t)) \\ + \frac{1}{K}\sum_{t=1}^{T}\sum_{\tau=0}^{K-1} \vec{W}^{\text{T}}[\vec{X}^{\text{RMHC}^{(\tau)}}(t) - \vec{X}^{\text{RMHC}^{(\tau)}}(t-1)]^+.\end{aligned}$$

Next, we can change the order of the summations over t and τ. Meanwhile, we put together all rounds of $\text{RMHC}^{(\tau)}$ from the same row of G^{old}. We can then get

$$\begin{aligned}\text{Cost}^{\text{ARMHC}}(1:\mathcal{T}) \leq \frac{1}{K}\sum_{\tau=0}^{K-1}\Bigg\{\sum_{u=-1}^{\lfloor\frac{\mathcal{T}}{K}\rfloor}\sum_{t=\tau+Ku}^{\tau+Ku+K-1}\Big[h_t(\vec{X}^{\text{RMHC}^{(\tau)}}(t)) \\ + \vec{W}^{\text{T}}[\vec{X}^{\text{RMHC}^{(\tau)}}(t) - \vec{X}^{\text{RMHC}^{(\tau)}}(t-1)]^+\Big]\Bigg\}.\end{aligned}$$

The rest of the proof follows the same steps from Eq.(37) to Eq.(38). Hence, for the online optimization problem (2) using convex service costs, the competitive ratio of ARMHC described in Sec. 3.4 is still $\frac{K+1}{K}$. □

G PROOF OF THEOREM 4.2

The proof again follows a similar line as in steps 1, 2 and 3 in the proof of Theorem 3.1 in Sec. 3.3. We first provide some preliminary results.

G.1 Preliminary Results

By introducing an auxiliary variable $\vec{Y}(t)$ for the switching-cost term in Eq.(41), we can get an equivalent formulation of the optimization problem (41) as follows.

$$\min_{\substack{\{\vec{X}(t:t+k-1),\\ \vec{Y}(t:t+k-1)\}}} \left\{\sum_{s=t}^{t+k-1}\rho(s-t)\vec{C}^{\text{T}}(s)\vec{X}(s) + \sum_{s=t}^{t+k-1}\vec{W}^{\text{T}}\vec{Y}(s) - \left[\vec{W}^{\text{T}} - \vec{C}^{\text{T}}(t+k)\right]^+ \cdot \rho(k-1)\vec{X}(t+k-1)\right\} \tag{77a}$$

$$\text{sub. to: } \boldsymbol{B}_1\vec{X}(t) \geq \boldsymbol{B}_2\vec{A}(t), \tag{77b}$$

$$\boldsymbol{B}_1\vec{X}(s) \geq \boldsymbol{B}_2\overline{\vec{A}}(s), \text{ for all time } s \in [t+1, t+k-1], \tag{77c}$$

$$\vec{Y}(t) \geq \rho(0)\vec{X}(t) - \rho(k-1)\vec{X}(t-1), \tag{77d}$$

$$\vec{Y}(s) \geq \rho(s-t)\vec{X}(s) - \rho(s-t-1)\vec{X}(s-1), \text{for all time } s \in [t+1, t+k-1], \tag{77e}$$

$$\vec{X}(s), \vec{Y}(s) \geq 0, \text{ for all time } s \in [t, t+k-1]. \tag{77f}$$

Step-1: For each round of GRMHC$^{(\tau)}$ from time t to $t+K-1$, we define the primal cost

$$\begin{aligned}&\text{Cost}^{\text{GRMHC}^{(\tau)}}(t:t+k-1) \triangleq \sum_{s=t}^{t+k-1} \vec{C}^{\text{T}}(t)\rho(s-t)\vec{X}^{\text{GRMHC}^{(\tau)}}(s)\\&+\vec{W}^{\text{T}}\left[\rho(0)\vec{X}^{\text{GRMHC}^{(\tau)}}(t)-\rho(k-1)\vec{X}^{\text{GRMHC}^{(\tau)}}(t-1)\right]^{+}\\&+\sum_{s=t+1}^{t+k-1}\vec{W}^{\text{T}}\left[\rho(s-t)\vec{X}^{\text{GRMHC}^{(\tau)}}(s)-\rho(s-t-1)\vec{X}^{\text{GRMHC}^{(\tau)}}(s-1)\right]^{+}.\end{aligned} \tag{78}$$

Next, we let $\vec{\beta}^{\text{GRMHC}^{(\tau)}}(t)$, $\vec{\beta}^{\text{GRMHC}^{(\tau)}}(t+1:t+k-1)$, $\vec{\theta}^{\text{GRMHC}^{(\tau)}}(t)$ and $\vec{\theta}^{\text{GRMHC}^{(\tau)}}(t+1:t+k-1)$ be the optimal dual solutions to the dual of the optimization problem (77). Then, for each $\tau \in [0, k-1]$, we can define the online dual cost in a similar manner as for RMHC$^{(\tau)}$ in Eq.(18), i.e., the online dual cost of GRMHC$^{(\tau)}$ is

$$\begin{aligned}&D^{\text{GRMHC}^{(\tau)}}(t:t+k-1|\overline{\vec{A}})\\&\triangleq\left(\vec{\beta}^{\text{GRMHC}^{(\tau)}}(t)\right)^{\text{T}}\boldsymbol{B}_2\vec{A}(t)+\sum_{s=t+1}^{t+k-1}\left(\vec{\beta}^{\text{GRMHC}^{(\tau)}}(s)\right)^{\text{T}}\boldsymbol{B}_2\overline{\vec{A}}(s).\end{aligned} \tag{79}$$

Further, we define the following two tail-terms in the same form of Eq.(19), except that now they are multiplied by the weight $\rho(k-1)$, i.e.,

$$\phi^{(\tau)}(t-1) \triangleq -\rho(k-1)\vec{W}\vec{X}^{\text{GRMHC}^{(\tau)}}(t-1), \tag{80a}$$

$$\psi^{(\tau)}(t+k-1) \triangleq \rho(k-1)[\vec{W}-\vec{C}(t+k)]^{+}\vec{X}^{\text{GRMHC}^{(\tau)}}(t+k-1). \tag{80b}$$

Then, we have Lemma G.1 below, which is similar to Lemma 3.2.

Lemma G.1. *(Per-round gap between the primal and dual costs of GRMHC$^{(\tau)}$) For each $\tau \in [0, k-1]$ and any $t = \tau + ku$ where $u = -1, 0, \ldots, \lfloor \frac{\mathcal{T}}{k} \rfloor$, we have*

$$\begin{aligned}&Cost^{GRMHC^{(\tau)}}(t:t+k-1) \le D^{GRMHC^{(\tau)}}\left(t:t+k-1|\overline{\vec{A}}\right)\\&\quad+\phi^{(\tau)}(t-1)+\psi^{(\tau)}(t+k-1),\ \textit{for all } \vec{A}(t:t+k-1).\end{aligned} \tag{81}$$

The proof follows the same line as the proof of Lemma 3.2 in Appendix C. There are two differences. First, we need to replace the real input $\vec{A}(t)$ with the upper bound $\overline{\vec{A}}(t)$ in Eq.(50a) for time $t+1$ to time $t+K-1$. This is because GRMHC considers the future uncertain hard constraints based on the upper bound of the future inputs. Second, as in Eq.(77), we need to add the same weight $\rho(0:k-1)$ to the corresponding decision variables of each equation during the proof.

Step-2: Similar to Step-2 in Sec. 3.3, here we can still organize the online dual cost in Eq.(79), i.e., $D^{\text{GRMHC}^{(\tau)}}\left(t:t+K-1|\overline{\vec{A}}\right)$, into the matrix G^{new} using the re-stitching idea. Second, as shown in Step-2 in Sec.3.3, we can add "0" for the gap in each row of G^{new}. However, since we add weights in the optimization problem (77), the online dual variables $\vec{\beta}^{\text{GRMHC}^{(\tau)}}(t:t+k-1)$ and $\vec{\theta}^{\text{GRMHC}^{(\tau)}}(t:t+k-1)$ may not be feasible for the offline dual optimization problem (14). Fortunately, from Eq.(77), we can verify that these dual variables from each version τ of GRMHC$^{(\tau)}$ satisfy

$$\begin{aligned}&\rho(s-t)\vec{C}(s)-\boldsymbol{B}_1^{\text{T}}\vec{\beta}^{\text{GRMHC}^{(\tau)}}(s)+\rho(s-t)\vec{\theta}^{\text{GRMHC}^{(\tau)}}(s)\\&\quad-\rho(s-t)\vec{\theta}^{\text{GRMHC}^{(\tau)}}(s+1)\ge 0,\ \text{for all time } s\in[t,t+k-2],\end{aligned} \tag{82a}$$

$$\rho(k-1)\vec{C}(t+k-1) - \boldsymbol{B}_1^{\mathrm{T}}\vec{\beta}^{\mathrm{GRMHC}^{(\tau)}}(t+k-1) + \rho(k-1)\vec{\theta}^{\mathrm{GRMHC}^{(\tau)}}(t+k-1)$$
$$- \rho(k-1)[\vec{W} - \vec{C}(t+k)]^+ \geq 0, \tag{82b}$$
$$\vec{W} - \vec{\theta}^{\mathrm{GRMHC}^{(\tau)}}(s) \geq 0, \text{ for all time } s \in [t, t+k-1], \tag{82c}$$

Further, these dual variables are all non-negative. Then, we can construct a set of dual variables that

$$\begin{cases} \tilde{\vec{\beta}}^{\mathrm{GRMHC}^{(\tau)}}(s) = \frac{\vec{\beta}^{\mathrm{GRMHC}^{(\tau)}}(s)}{\rho(s-t)}, \text{ for all time } s \in [t, t+k-1], \\ \tilde{\vec{\theta}}^{\mathrm{GRMHC}^{(\tau)}}(s)) = \vec{\theta}^{\mathrm{GRMHC}^{(\tau)}}(s)), \text{ for all time } s \in [t, t+k-1]. \end{cases} \tag{83}$$

We can then define the online dual cost

$$\tilde{D}^{\mathrm{GRMHC}^{(\tau)}}(t : t+k-1|\vec{A}) \triangleq \sum_{s=t}^{t+k-1} \left(\tilde{\vec{\beta}}^{\mathrm{GRMHC}^{(\tau)}}(s)\right)^{\mathrm{T}} \boldsymbol{B}_2\vec{A}(s). \tag{84}$$

Notice that the feasible dual variables and the real inputs are used in Eq.(84). Applying the re-stitching idea, we can organize these online dual costs into the matrix G^{new}. Meanwhile, in the i-th row of G^{new}, we still introduce

$$\begin{cases} \tilde{\vec{\beta}}(i+(K+1)u+K) = \vec{0}, \\ \tilde{\vec{\theta}}(i+(K+1)u+K) = [\vec{W} - \vec{C}(i+(K+1)u+K)]^+, \end{cases}$$

at time $i+(K+1)u+K$ for $u = -1, 0, \ldots, \left\lfloor \frac{\mathcal{T}}{K+1} \right\rfloor$. Then, Lemma 3.3 still holds. The proof is the same as the proof in Appendix D. The only difference is that now we need to replace the dual variables there by the corresponding dual variables in Eq.(83).

Step-3: Notice that the two tail-terms in Eq.(80) are in the same form as the tail-terms in Eq.(19), except that each of them is multiplied by the same weight $\rho(k-1)$. Multiply the tail-terms during the proof in Appendix E by the same weight, each step still holds. Thus, the sum of the tail-terms from all rounds of GRMHC are still upper-bounded by 0. Therefore, Lemma 3.4 still holds.

G.2 Proof of Theorem 4.2

Proof. First, from the cost definition in Eq.(4), we know that the total cost of WRMHC is

$$\mathrm{Cost}^{\mathrm{WRMHC}}(1:\mathcal{T}) = \sum_{t=1}^{\mathcal{T}} \vec{C}^{\mathrm{T}}(t)\vec{X}^{\mathrm{WRMHC}}(t) + \sum_{t=1}^{\mathcal{T}} \vec{W}^{\mathrm{T}}[\vec{X}^{\mathrm{WRMHC}}(t) - \vec{X}^{\mathrm{WRMHC}}(t-1)]^+.$$

Then, in Eq.(42), we notice that WRMHC takes the weighted average of K versions of GRMHC decisions. Recall the notation introduced in Sec. 4.2 that $\tilde{\tau}(t,i) \triangleq (t-i) \bmod k$ for any $i \in [0, k-1]$. Then, we have

$$\mathrm{Cost}^{\mathrm{WRMHC}}(1:\mathcal{T}) = \sum_{t=1}^{\mathcal{T}} \left\{ \vec{C}^{\mathrm{T}}(t) \cdot \frac{\sum_{i=0}^{k-1} \rho(i)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,i))}}(t)}{\sum_{i=0}^{k-1} \rho(i)} \right\}$$
$$+ \sum_{t=1}^{\mathcal{T}} \vec{W}^{\mathrm{T}} \cdot \left[\frac{\sum_{i=0}^{k-1} \rho(i)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,i))}}(t)}{\sum_{i=0}^{k-1} \rho(i)} - \frac{\sum_{i=0}^{k-1} \rho(i)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t-1,i))}}(t-1)}{\sum_{i=0}^{k-1} \rho(i)} \right]^+.$$

Since $\tilde{\tau}(t-1,i) = \tilde{\tau}(t,i+1)$, we have

$$\sum_{i=0}^{k-1} \rho(i)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t-1,i))}}(t-1) = \sum_{i=0}^{k-1} \rho(i)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,i+1))}}(t-1) \tag{86a}$$

$$= \sum_{i=0}^{k-2} \rho(i)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,i+1))}}(t-1) + \rho(k-1)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,k))}}(t-1) \tag{86b}$$

$$= \sum_{i=1}^{k-1} \rho(i-1)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,i))}}(t-1) + \rho(k-1)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,0))}}(t-1). \tag{86c}$$

Here, we obtain Eq.(86b) by simply rewriting the summation in Eq.(86a) and take the last term out. Eq.(86c) is true because $\tilde{\tau}(t,k) = \tilde{\tau}(t,0)$. Then, due to the convexity of the cost terms and Jensen's Inequality, we have

$$\begin{aligned}\mathrm{Cost}^{\mathrm{WRMHC}}(1:\mathcal{T}) \leq & \frac{1}{\sum_{i=0}^{k-1}\rho(i)} \sum_{t=1}^{\mathcal{T}} \Bigg\{ \sum_{i=0}^{k-1} \vec{C}(t)\rho(i)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,i))}}(t) \\ & + \sum_{i=1}^{k-1} \vec{W}\left[\rho(i)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,i))}}(t) - \rho(i-1)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,i))}}(t-1)\right]^+ \\ & + \vec{W}\left[\rho(0)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,0))}}(t) - \rho(k-1)\vec{X}^{\mathrm{GRMHC}^{(\tilde{\tau}(t,0))}}(t-1)\right]^+ \Bigg\}.\end{aligned}$$

Next, we change the order of the summations over t and i. Meanwhile, we put together all rounds of $\mathrm{GRMHC}^{(\tau)}$ from the same row of G^{old}. We can then get

$$\begin{aligned}&\mathrm{Cost}^{\mathrm{WRMHC}}(1:\mathcal{T}) \\ &\leq \frac{1}{\sum_{i=0}^{k-1}\rho(i)} \sum_{\tau=0}^{k-1} \sum_{u=-1}^{\lfloor \frac{\mathcal{T}}{k} \rfloor} \Bigg\{ \sum_{t=\tau+ku}^{\tau+ku+k-1} \vec{C}^{\mathrm{T}}(t)\rho(t-\tau-ku)\vec{X}^{\mathrm{GRMHC}^{(\tau)}}(t) \\ &\quad + \sum_{t=\tau+ku+1}^{\tau+ku+k-1} \vec{W}^{\mathrm{T}}\left[\rho(t-\tau-ku)\vec{X}^{\mathrm{GRMHC}^{(\tau)}}(t) - \rho(t-\tau-ku-1)\vec{X}^{\mathrm{GRMHC}^{(\tau)}}(t-1)\right]^+ \\ &\quad + \vec{W}^{\mathrm{T}}\left[\rho(0)\vec{X}^{\mathrm{GRMHC}^{(\tau)}}(\tau+ku) - \rho(k-1)\vec{X}^{\mathrm{GRMHC}^{(\tau)}}(\tau+ku-1)\right]^+ \Bigg\}.\end{aligned} \tag{88}$$

Then, applying Lemma G.1, we have

$$\begin{aligned}&\mathrm{Cost}^{\mathrm{WRMHC}}(1:\mathcal{T}) \\ &\leq \frac{1}{\sum_{i=0}^{k-1}\rho(i)} \sum_{\tau=0}^{k-1} \sum_{u=-1}^{\lfloor \frac{\mathcal{T}}{k} \rfloor} \Big\{ D^{\mathrm{GRMHC}^{(\tau)}}(\tau+ku:\tau+ku+k-1|\overline{\vec{A}}) \\ &\quad + \phi^{(\tau)}(\tau+ku-1) + \psi^{(\tau)}(\tau+ku+k-1) \Big\}.\end{aligned} \tag{89}$$

Notice that the online dual costs defined in Eq.(79) is based on the upper bound of the future inputs $\overline{\vec{A}}(t+1:t+k-1)$. Moreover, the corresponding dual variables may not be feasible to the offline optimization problem (13). Fortunately, in Eq.(84), we construct the online dual costs with feasible dual variables and the real inputs. To relate the online dual costs defined in Eq.(79) with the online dual costs defined in Eq.(84), we have

$$
\begin{aligned}
&D^{\mathrm{GRMHC}^{(\tau)}}(t:t+k-1|\overline{\vec{A}}) \\
&= \sum_{s=t}^{t+k-1} \left(\tilde{\vec{\beta}}^{\mathrm{GRMHC}^{(\tau)}}(s)\right)^{\mathrm{T}} \boldsymbol{B}_2 \vec{A}(s) \frac{\rho(s-t)\overline{\vec{A}}(s)}{\vec{A}(s)} && \text{(90a)} \\
&\leq \sum_{s=t}^{t+k-1} \left(\tilde{\vec{\beta}}^{\mathrm{GRMHC}^{(\tau)}}(s)\right)^{\mathrm{T}} \boldsymbol{B}_2 \vec{A}(s) \rho(s-t) \max_{m\in[1,M]} \frac{\overline{a}_m(s)}{a_m(s)} && \text{(90b)} \\
&\leq \sum_{s=t}^{t+k-1} \left(\tilde{\vec{\beta}}^{\mathrm{GRMHC}^{(\tau)}}(s)\right)^{\mathrm{T}} \boldsymbol{B}_2 \vec{A}(s) \rho(s-t) \max_{m\in[1,M]} \frac{f_m^{\mathrm{up}}(s-t)}{f_m^{\mathrm{low}}(s-t)} && \text{(90c)} \\
&\leq D^{\mathrm{GRMHC}^{(\tau)}}(t:t+k-1|\vec{A}) \max_{\substack{\{i\in[0,k-1]\\ m\in[1,M]\}}} \rho(i) \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)}, && \text{(90d)}
\end{aligned}
$$

Here, the multiplications and divisions of the input vector $\vec{A}(t)$ in Eq.(90a) are entry-wise. As mentioned in Sec. 2.2, in Eqs.(90c) and (90d), we have $\frac{f_m^{\mathrm{up}}(0)}{f_m^{\mathrm{low}}(0)} = 1$ for all $m \in [1, M]$. Then, for the terms involving the online dual cost $D^{\mathrm{GRMHC}^{(\tau)}}(\tau+Ku:\tau+Ku+K-1|\overline{\vec{A}})$ in Eq.(89), applying Eq.(90) and the re-stitching idea, we have,

$$
\begin{aligned}
&\sum_{\tau=0}^{k-1} \sum_{u=-1}^{\lfloor \frac{\mathcal{T}}{k} \rfloor} \left\{ D^{\mathrm{GRMHC}^{(\tau)}}(\tau+ku:\tau+ku+k-1|\overline{\vec{A}}) \right\} \\
&\leq \sum_{\tau=0}^{k-1} \sum_{u=-1}^{\lfloor \frac{\mathcal{T}}{k} \rfloor} D^{\mathrm{GRMHC}^{(\tau)}}(\tau+ku:\tau+ku+k-1|\vec{A}) \cdot \max_{\substack{\{i\in[0,k-1]\\ m\in[1,M]\}}} \rho(i) \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)} && \text{(91a)} \\
&\leq \sum_{i=1}^{k} \sum_{j=1}^{\lfloor \frac{\mathcal{T}}{k} \rfloor+2} G_{i,j}^{\mathrm{old}} \max_{\{i\in[0,k-1] m\in[1,M]\}} \rho(i) \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)} && \text{(91b)} \\
&\leq \sum_{i=1}^{k+1} \sum_{j=1}^{2\lfloor \frac{\mathcal{T}}{k+1} \rfloor+3} G_{i,j}^{\mathrm{new}} \max_{\{i\in[0,k-1], m\in[1,M]\}} \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)} \rho(i) && \text{(91c)} \\
&\leq (k+1) D^{\mathrm{OPT}}(1:\mathcal{T}) \cdot \max_{\{i\in[0,k-1], m\in[1,M]\}} \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)} \rho(i). && \text{(91d)}
\end{aligned}
$$

Therefore, combining Eqs.(89), (91), we have that the total cost of WRMHC satisfies

$$
\begin{aligned}
&\mathrm{Cost}^{\mathrm{WRMHC}}(1:\mathcal{T}) \\
&\leq \frac{k+1}{\sum_{i=0}^{k-1} \rho(i)} D^{\mathrm{OPT}}(1:\mathcal{T}) \cdot \max_{\{i\in[0,k-1], m\in[1,M]\}} \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)} \rho(i)
\end{aligned}
$$

$$\leq \mathrm{Cost}^{\mathrm{OPT}}(1:\mathcal{T}) \cdot \frac{k+1}{\sum_{i=0}^{k-1} \rho(i)} \cdot \max_{\{i\in[0,k-1],\, m\in[1,M]\}} \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)} \rho(i).$$

□

H PROOF OF LEMMA 4.3

Proof. The key to prove Lemma 4.3 is to solve the following optimization problem, given $k \in [1, K]$,

$$\min_{\rho(0:k-1)\geq 0} \frac{\max_{\{i\in[0,k-1],\, m\in[1,M]\}} \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)} \rho(i)}{\sum_{i=0}^{k-1} \rho(i)}. \tag{93}$$

Suppose that the optimal solution to Eq.(93) is $\tilde{\rho}(0:k-1)$. We let

$$\alpha \triangleq \max_{\{i\in[0,k-1],\, m\in[1,M]\}} \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)} \tilde{\rho}(i). \tag{94}$$

Thus, from Eq.(94), we have

$$\frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)} \tilde{\rho}(i) \leq \alpha, \text{ for all } i \in [0, k-1] \text{ and } m \in [1, M] \tag{95a}$$

$$\Rightarrow \tilde{\rho}(i) \leq \frac{\alpha f_m^{\mathrm{low}}(i)}{f_m^{\mathrm{up}}(i)}, \text{ for all } i \in [1, k-1] \text{ and } m \in [1, M] \tag{95b}$$

$$\Rightarrow \begin{cases} \tilde{\rho}(0) \leq \alpha, \\ \tilde{\rho}(i) \leq \min_{m\in[1,M]} \frac{\alpha f_m^{\mathrm{low}}(i)}{f_m^{\mathrm{up}}(i)}, \text{ for all } i \in [1, k-1]. \end{cases} \tag{95c}$$

Then, the objective function of the outer minimization in Eq.(93) satisfies

$$\frac{\max_{\{i\in[1,k-1],\, m\in[1,M]\}} \frac{f_m^{\mathrm{up}}(i)}{f_m^{\mathrm{low}}(i)} \rho(i)}{\sum_{i=0}^{k-1} \rho(i)} \tag{96a}$$

$$\geq \frac{\alpha}{\alpha + \sum_{i=1}^{k-1} \min_{m\in[1,M]} \frac{\alpha f_m^{\mathrm{low}}(i)}{f_m^{\mathrm{up}}(i)}} \tag{96b}$$

$$= \frac{1}{1 + \sum_{i=1}^{k-1} \min_{m\in[1,M]} \frac{f_m^{\mathrm{low}}(i)}{f_m^{\mathrm{up}}(i)}}. \tag{96c}$$

In other words, (96c) corresponds a lower bound on the optimal value of (93). On the other hand, by setting

$$\begin{cases} \tilde{\rho}(0) = 1, \\ \tilde{\rho}(i) = \min_{m\in[1,M]} \frac{f_m^{\mathrm{low}}(i)}{f_m^{\mathrm{up}}(i)}, \text{ for all } i \in [1, k-1], \end{cases} \tag{97}$$

the objective value in the outer minimization in Eq.(93) is exactly (96c). Hence, the weights in Eq.(97) is the solution to (93). □

Received February 2019; revised March 2019; accepted April 2019