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

ABSTRACT
While using look-ahead information is known to improve the competitive ratio 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 potentially can 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 input in a future look-ahead window of size $K$, we propose an Averaging Regularized Moving Horizon Control (ARMHC) algorithm that can achieve a competitive ratio of $\frac{K}{K+1}$. To the best of our knowledge, ARMHC is the first to attain a low competitive ratio that is independent of either the switching-cost and service-cost coefficients, or the maximum and minimum size of the input. Then, for the case when the future look-ahead has errors, we propose 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 hard constraints. Finally, our analysis extends online primal-dual analysis to the case with look-ahead using a novel "re-stitching" idea, which is of independent interest.

CCS CONCEPTS
• Theory of computation → Online algorithms; Convex optimization; • Mathematics of computing → Mathematical analysis;

KEYWORDS
Online convex optimization (OCO), look-ahead, competitive analysis, online primal-dual analysis

ACM Reference Format:

1 INTRODUCTION
In this paper, we study online convex optimization (OCO) problems with switching costs and hard constraints [2, 11, 18]. In its generic form, an OCO problem proceeds in times. At each time $t$, the environment (or adversary) reveals the input $\hat{A}(t) \in \mathbb{R}^{M \times 1}$. Then, the decision maker chooses the decision $\hat{X}(t) \in \mathbb{R}^{N \times 1}$, which is based on all revealed inputs from $\hat{A}(1)$ to $\hat{A}(t)$. A service cost $c_t(\hat{X}(t), \hat{A}(t))$ is incurred for the decision $\hat{X}(t)$. Further, there is a switching cost that penalizes the increments $|\hat{X}(t) - \hat{X}(t-1)|_1$ for each time $t$, where $|x|_1 = \max \{x, 0\}$. In the majority of this paper, we focus on the case where the service cost is linear in $\hat{X}(t)$ and the switching cost is linear in the increment $|\hat{X}(t) - \hat{X}(t-1)|_1$, although some of the results will also generalized to convex service costs. Such an OCO formulation can be used to model online decisions in many application domains, such as machine learning [8, 17, 20, 23], networking [11, 12, 14, 18], cloud computing [4, 5, 13, 19], cyber-physical systems [10, 15] and wireless systems [9, 16]. For example, in the Network Functions Virtualization (NFV) orchestration and scaling problems, the service cost can model the setup cost and distance cost. The switching cost can model the overhead for migrating demand among different Virtualized Network Function (VNF) instances within or across data centers [6, 18]. In many practical settings, the decision $\hat{X}(t)$ is subject to hard constraints that must be satisfied at each time $t$. For example, the incoming traffic must be served by VNFs placed in the system [18]. Consider a time-horizon $T$. In an offline setting, the entire input sequence $\hat{A}(1)$ to $\hat{A}(T)$ 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 $\hat{A}(t)$ 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 [2, 4, 5, 18], it remains a challenge to design online algorithms with low competitive ratios. For example, the regularization method can be used on very general classes of OCO problems without any future information [2]. However, the resulting competitive ratio depends on the upper bound $\max \{\|\hat{A}(t)\|_1\}$ and the lower bound $\min \{\|\hat{A}(t)\|_1\}$ of the input trajectory$^1$, and hence can be quite large in practice. This is not surprising because, when there is absolutely no future knowledge, the competitive ratio often has to be large due to arbitrarily adversarial inputs. Fortunately, in many applications, such as resource allocation in data centers [11, 18], power dispatching in smart grid [21, 22] and video streaming in wireless systems [9, 16], the future input can be predicted to some extent. Such future information can be modeled with a look-ahead window [11]. Intuitively, the further ahead such prediction is provided, 

$^1$For a vector $\hat{A} = (a_1, a_2, \ldots, a_M)$, the infinity norm $\|\hat{A}\|_\infty = \max_{m=1,2,\ldots,M} |a_m|$. 

---

Permission to make digital or hard copies of part or all 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 third-party components of this work must be honored. For all other uses, contact the owner/author(s). POMACS’19, June 2019, Phoenix, Arizona, USA © 2019 Copyright held by the owner/author(s). ACM ISBN 123-4567-24-567/08/06. https://doi.org/10.475/123_4
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.

There has been a number of recent studies on using look-ahead to improve online decisions. Nonetheless, these existing results are still unsatisfactory due to potentially high competitive ratios. Receding Horizon Control (RHC), also known as Model Predictive Control (MPC), is a popular framework in the literature to use look-ahead [3, 11]. 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 size of the look-ahead window $K$ increases [5, 11]. Recently, Lin et al. propose the Averaging Fixed Horizon Control (AFHC) algorithm, which averages the decisions from multiple rounds of Fixed Horizon Control (FHC). AFHC is shown to be $(1 + \max\frac{w_n}{(K+1)c_n})$-competitive, where $c_n$ is the service-cost coefficient and $w_n$ is the switching-cost coefficient [11]. Note that while this competitive ratio of AFHC improves as $K$ increases, it still increases as $\max\frac{w_n}{c_n}$ increases. In contrast, note that the competitive ratios of the regularization method [2] does not depend on $\frac{w_n}{c_n}$. Therefore, it remains an open question whether one can develop an online algorithm that can achieve a low competitive ratio independent of $w_n$ and $c_n$ even under perfect look-ahead.

For imperfect look-ahead, existing results mainly focus on the competitive difference of RHC, AFHC, and CHC (Committed Horizon Control) [5]. Note that competitive difference is the worst-case difference between the cost of an online algorithm and that of the optimal offline solution, while the competitive ratio is the worst-case 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 [5] still depends on such problem parameters as the coefficients of the switching costs. Moreover, none of these results consider hard constraints. Recently, Shi et al. propose a Robust Affine Policy (RAP) that optimizes the competitive ratio for OCO problems within the class of affine policies [18]. Although RAP can work with hard constraints, its competitive ratio still depends on the switching-cost coefficient $w_n$ [18]. Thus, it remains an open problem to develop online algorithms with low competitive ratios under imperfect look-ahead and hard constraints.

In summary, none of the existing results can achieve low competitive ratios independent of the switching-cost coefficients and the service-cost coefficients under either perfect look-ahead or imperfect look-ahead. In this paper, we design new online algorithms for online convex optimization (OCO) problems with switching costs and hard constraints. Our proposed algorithms can significantly improved competitive ratios for both perfect look-ahead and imperfect look-ahead.

(1) First, we study the case with perfect look-ahead, i.e., at time $t$, the inputs from $\tilde{A}(t)$ to $\tilde{A}(t+K)$ are given to the decision maker. We first develop a novel Averaging Regularized Moving Horizon Control (ARMHC) algorithm that is $(1 + \frac{w_n}{Kc_n})$-competitive. 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 such as the coefficients of the switching costs or the gap between 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 [2]. This suggests that even a small window of perfect look-ahead can be highly effective in reducing the competitive ratio. As $K$ 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. Finally, our proof carefully combines online primal-dual competitive analysis with a novel “re-stitching” idea to deal with the gap between look-ahead horizons, which is also of independent interest.

(2) Second, we study the case with imperfect look-ahead, i.e., at time $t$, only a possible trajectory of near-term future inputs from $\tilde{A}(t+1)$ to $\tilde{A}(t+K)$ is given to the decision maker, as well as upper and lower bounds of the possible future inputs. Our proposed variant of RHC will use the upper bounds of the future inputs in the decision, 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 the 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 a lower competitive ratio under imperfect look-ahead that is less sensitive to the gap between the upper and lower bounds of prediction. Finally, the competitive ratio of WRMHC is also independent of the switching-cost coefficient.

Besides, contrary to the existing algorithms using look-ahead, WRMHC can ensure that hard constraints are satisfied at all times. This is achieved by using the upper bound of the possible future inputs in computing Generalized Regularized Moving Horizon Control (GRMHC) decisions in each look-ahead window.

2 PROBLEM FORMULATION

In this section, we provide the formulation of the online convex optimization (OCO) problems 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 $\tilde{A}(t) = [a_{m}, m = 1, ..., M]^{T} \in \mathbb{R}^{M \times 1}$ are revealed ($[ ]^{T}$ denotes the transpose of a vector or matrix). Then, the decision maker chooses a decision $\tilde{X}(t) = [x_{n}(t), n = 1, ..., N]^{T} \in \mathbb{R}^{N \times 1}$ from the convex decision set $\mathcal{X}_{t}(\tilde{A}(t))$. The set $\mathcal{X}_{t}(\tilde{A}(t))$ may contain
hard constraints, which can be written as linear inequalities in $(\tilde{X}(t), \tilde{A}(t))$, i.e.,

$$B_1 \tilde{X}(t) \geq B_2 \tilde{A}(t), \text{ for all time } t = 1, \ldots, T,$$

(1)

where $B_1$ is an $L \times N$ matrix and $B_2$ is an $L \times M$ matrix. We assume that $B_1$, $B_2 \geq 0$, where “$\geq$” is component-wise. For example, if each element of $B_1$ and $B_2$ is 1, then Eq.(1) becomes $\sum_{n=1}^{N} x_n(t) \geq \sum_{m=1}^{M} a_m(t)$, for all time $t$. It can be used to model the hard constraint that the incoming traffic must be served by the servers in data centers [11, 18] or the power demand must be satisfied by the dispatching decisions in smart grids [21].

Provided that the hard constraint is met at time $t$, a service cost is incurred. For the moment, we assume that the service cost is linear, i.e., is given by $\tilde{C}^T(t)\tilde{X}(t)$, where $\tilde{C}(t) = [c_n(t), n = 1, \ldots, N]^T \in \mathbb{R}^{N \times 1}$ is the service-cost coefficient. (Our results for the perfect look-ahead case can also be generalized to convex service costs. See Sec. 3.3.) Note that if the hard constraint is not satisfied, we assume that the service cost will be $+\infty$. Additionally, there is a switching cost $\tilde{W}^T(t)\tilde{X}(t) - \tilde{X}(t - 1)^*$ that penalizes the increment of each entry of the decision $\tilde{X}(t)$ at time $t$, where $\tilde{W} = [w_m, n = 1, \ldots, N]^T \in \mathbb{R}^{N \times 1}$ is the switching-cost coefficient.

We let $[l_1, l_2]$ denote the set $\{l_1, l_1 + 1, \ldots, l_2\}$. If all the future inputs were known, then this problem would have been a standard convex optimization problem as follows,

$$\begin{align*}
\min_{\{\tilde{X}(1), \ldots, \tilde{X}(T)\}} & \left\{ \sum_{t=1}^{T} \tilde{C}^T(t)\tilde{X}(t) + \sum_{t=1}^{T} \tilde{W}^T(\tilde{X}(t) - \tilde{X}(t - 1))^* \right\} \\
\text{sub. to:} & \quad B_1 \tilde{X}(t) \geq B_2 \tilde{A}(t), \text{ for all time } t \in [1, T], \\
& \quad \tilde{X}(t) \geq 0, \text{ for all time } t \in [1, T],
\end{align*}$$

(2a)

where $\tilde{X}(0) = 0$ as typically assumed in the literature of OCO problems [5].

However, as in [5, 8, 17, 20, 23], 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 $\tilde{X}(t)$ based on only the current input $\tilde{A}(t)$ and a limited amount of look-ahead information for future inputs (which will be defined precisely shortly). Moreover, once the decision $\tilde{X}(t)$ is made, the service cost $\tilde{C}^T(t)\tilde{X}(t)$ and the switching cost $\tilde{W}^T(\tilde{X}(t) - \tilde{X}(t - 1))^*$ are incurred. The decision made is irrevocable, which means in the future time-slots $t + 1, t + 2, \ldots, T$, she cannot go back and revise the decision $\tilde{X}(t)$. Thus, this problem becomes an online problem.

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

### 2.2 Modeling Look-Ahead

As we discussed in the introduction, in many applications the future input can be predicted to some extent. We can model such partial future information via a look-ahead window. Let $K, K \geq 1$, denote the size of the look-ahead window. At each time $t$, the decision maker will not only know the exact input $\tilde{A}(t)$ of time $t$, but also know some information about the future inputs $\tilde{A}(t + 1), \tilde{A}(t + 2), \ldots, \tilde{A}(t + K)$. Note that the future inputs beyond $t + K + 1$ 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 $\tilde{C}(t + 1 : t + K)$ and inputs $\tilde{A}(t + 1 : t + K)$ within the look-ahead window. Thus, the online algorithm can make the current decision $\tilde{X}(t)$ based on not only all revealed service-cost coefficients $\tilde{C}(1 : t)$, inputs $\tilde{A}(1 : t)$ and past decisions $\tilde{X}(t - 1 : t)$, but also future values of $\tilde{C}(t + 1 : t + K)$ and $\tilde{A}(t + 1 : t + K)$ in the look-ahead window. Thus, as in [5, 8, 17, 20, 23], we focus on the setting where $\tilde{A}(1 : t + K)$ even in the look-ahead window. Instead, she is given a predicted trajectory $\tilde{A}_{\text{pred}}(t + 1 : t + K) = \{\tilde{a}_{t,m}(i + t), m \in [1, M], i \in [1, K]\}$ from time $t + 1$ to $t + K$, as well as bounds on how far the real near-term future inputs $\tilde{A}(t + 1 : t + K)$ can deviate from the predicted trajectory. Specifically, at a future time $t + i, i = 1, 2, \ldots, K$, the ratio of the $m$-th component of the real near-term future input to that of the predicted input is upper-bounded by

$$\frac{a_{m}(t + i)}{\tilde{a}_{t,m}(i + t)} \leq \frac{f_{\text{up}}(i)}{f_{\text{up}}(i)} \leq \frac{f_{\text{low}}(i)}{f_{\text{low}}(i)}$$

(3)

and lower-bounded by

$$\frac{a_{m}(t + i)}{\tilde{a}_{t,m}(i + t)} \geq \frac{f_{\text{low}}(i)}{f_{\text{low}}(i)} \leq \frac{f_{\text{up}}(i)}{f_{\text{up}}(i)}$$

(4)

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

Remark 1. The assumption that the service-cost coefficients $\tilde{C}(t + 1 : t + K)$ is still known in the imperfect look-ahead case is reasonable, e.g., this could due to the reason that the service-cost coefficients are fixed for the entire time-horizon in advance, e.g., the distance costs or setup costs are known in data centers [18].

### 2.3 The Performance Metric

As we shown in Eq.(2), the total cost received by an online algorithm $\pi$ over $T$ time slots will also includes both service costs and switching costs and is given by Eq.(3) as follows,

$$\text{Cost}^{\pi}(1 : T) = \sum_{t=1}^{T} \left( \tilde{C}^T(t)\tilde{X}^{\pi}(t) + \tilde{W}^T(\tilde{X}^{\pi}(t) - \tilde{X}^{\pi}(t - 1))^* \right),$$

(5)
where $\tilde{X}(t)$ is the decision made by the online algorithm $\pi$ at each time $t$. Let $\tilde{X}^{\text{OPT}}(1 : T)$ be the optimal offline solution to the optimization problem in Eq.(2), whose total cost is $\text{Cost}^{\text{OPT}}(1 : T)$. We evaluate an online algorithm $\pi$ using the competitive ratio $\text{CR}_\pi$, i.e., the worst-case ratio of its total cost to that of the optimal offline solution,

$$\text{CR}_\pi \triangleq \max_{\{\text{all possible } \tilde{A}(1 : T)\}} \frac{\text{Cost}_\pi(1 : T)}{\text{Cost}^{\text{OPT}}(1 : T)}.$$  \hspace{1cm} (6)

Thus, in this paper we are interested in developing algorithms with low competitive ratios.

### 3. THE PERFECT LOOK-AHEAD CASE

In this section, we consider first the case with perfect look-ahead. In [11], Averaging Fixed Horizon Control (AFHC) algorithm has been proposed as a way to utilize such perfect look-ahead information. It was shown in [11] that AFHC can achieve much smaller competitive ratios than Receding Horizon Control (RHC) algorithm 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 for general OCO problems. This Counter-example thus motivates us to develop new competitive online algorithm for perfect look-ahead case with significantly better competitive ratios.

### 3.1 A Counter-Example for AFHC

We first describe the behavior of AFHC from [11]. An AFHC algorithm with 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 $\tau$ be an integer from 0 to $K$. The $\tau$-th version of FHC, denoted by $\text{FHC}^{(\tau)}$, calculates decisions at time $t = \tau + (K + 1) \cdot u$, $u = -1, 0, ..., \left\lfloor \frac{T}{K+1} \right\rfloor$. Specifically, at each time $t = \tau + (K + 1) \cdot u$, $u = -1, 0, ..., \left\lfloor \frac{T}{K+1} \right\rfloor$, the current input $\tilde{A}(t)$ is revealed and inputs in the current look-ahead window $t + 1, ..., t + K$ are given. $\text{FHC}^{(\tau)}$ then calculates the solution to the following problem $Q^{\text{FHC}^{(\tau)}}(t)$:

$$\min_{\tilde{X}(t+K)} \left\{ \sum_{s=1}^{t+K} \left[ \tilde{C}^T(s) \tilde{X}(s) + \tilde{W}^T(s) [\tilde{X}(s) - \tilde{X}(s-1)]^T \right] \right\}$$

subject to: $B_1 \tilde{X}(s) \geq B_2 \tilde{A}(s)$, for all time $s \in [t, t + K]$,

$$\tilde{X}(s) \geq 0, \text{ for all time } s \in [t, t + K].$$

(7)

Here, the initial value $\tilde{X}(t - 1)$ is also from $\text{FHC}^{(\tau)}$, but is computed at an earlier time of $t - K - 1$, 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 $[1, T]$, we use the convention that $\tilde{A}(s) = 0$ and $\tilde{X}(s) = 0$ for all time $s \leq 0$ and $s > T$. By concatenating multiple rounds of $\text{FHC}^{(\tau)}$ at time $t = \tau + (K + 1) \cdot u$, $u = -1, 0, ..., \left\lfloor \frac{T}{K+1} \right\rfloor$, we get a unique decision sequence denoted by $\tilde{X}^{\text{FHC}^{(\tau)}}(1 : T)$. Then, AFHC simply takes the average of $\tilde{X}^{\text{FHC}^{(\tau)}}(1 : T)$ for all $\tau \in [0, K]$. The details are in [5] and also given in Algorithm 3 in Appendix A for completeness.

Intuitively, each $\text{FHC}^{(\tau)}$ is likely to make good decisions inside each look-ahead window because near-term future inputs are already known. However, the decision of $\text{FHC}^{(\tau)}$ at the end of a look-ahead window may still be poor. By averaging over multiple $\text{FHC}^{(\tau)}$, $\tau = 0, 1, ..., K$, AFHC ensures that such a poor “end-of-look-ahead” decision only occurs for one $\text{FHC}^{(\tau)}$ at a time. This is the key reason why AFHC can achieve a competitive ratio that decreases with $K$. (In contrast, the competitive ratio of RHC or FHC may not decrease with $K$.) However, with respect to other parameters, such as the switching-cost coefficient $w_0$, the effect of the above “end-of-look-ahead” decision can still be quite significant, especially when $K$ is not very large. In the following example, we demonstrate that AFHC may still 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 $\tilde{X}(t)$ and $\tilde{A}(t)$ are 1-dimensional, and thus can be replaced by scalars $x(t)$ and $a(t)$. 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 Counter-example, the sequence of the input at $a(j - 3) = a(j - 2) = \overline{a}$ and $a(j - 1) = a(j) = \underline{a}$, $j = 1, 2, ..., \frac{T}{4}$, which is plotted in Fig. 1a for $\overline{a} = 1000$ and $\underline{a} = 0$. The hard constraint is $x(t) \geq a(t)$ for all time $t$. For ease of explanation, we assume $\overline{a}$ is a multiple of 4 in this example. For other values of $\overline{a}$, this example can be easily adjusted, while the conclusion still holds.

Note that if $a(t) = \overline{a}$, then the decision maker will trivially set $x(t) = \overline{a}$. However, when $a(t) = \underline{a}$, the decision maker faces the dilemma between choosing a larger value $x(t) = \overline{a}$ (which incurs a high service cost but possibly lower switching cost for the future), or choosing a smaller value $x(t) < \overline{a}$ (which incurs a lower service cost but possibly higher switching cost for the future). In the following, we assume that the switching-cost coefficient $w$ is much larger than the service-cost coefficient $e(t)$.

**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 $x^{\text{OPT}}(t) = \overline{a}$ for all time $t$ (even when $a(t) = \underline{a}$) to avoid switching costs altogether. The decisions are plotted in Fig. 1b for $\overline{a} = 1000$ and $\underline{a} = 0$. The corresponding total...
cost is \(\text{Cost}^{\text{OPT}}(1 : T) = \sum_{t=1}^{T} c(t)\bar{a} + w\bar{a}.\) (The second term is the only switching cost that is incurred at time 1.)

**Cost of AFHC:** Since AFHC does not have all the future information, it is difficult for AFHC to emulate the optimal offline decisions.

To see this, note that as \(K = 3\), there are 4 versions of FH. The decisions of FH(0), FH(1), FH(2) and FH(3) are plotted in Fig. 1d for \(\bar{a} = 1000\) and \(q = 0\). Although, FH(2) and FH(3) follow exactly the same decisions as the optimal offline solution, FH(0) and FH(1) will have a different behavior due to the "end-of-look-ahead" problem that we mentioned before. For example, in the round from time \(t = 1\) to \(t = 4\), FH(1) sees input sequence \((\bar{a}, \bar{a}, q, q)\). FH(1) will give the decision sequence \((\bar{a}, \bar{a}, q, q)\), since there is no need to maintain a high decision \(x(1) = \bar{a}\) at time \(t = 3\) and AFHC does not know whether there exist large input \(\bar{a}\) beyond time 4. Please see Appendix A for the derivation of the decision sequences of different versions of FH.

Taking the average of these decisions, AFHC will gives the final decision sequence \((\bar{a}, \bar{a}, \bar{a}, q, q)\) for every round starting from time 1 and it is plotted in Fig. 1c for \(\bar{a} = 1000\) and \(q = 0\). (See Appendix A for details.) The corresponding total cost is \(\text{Cost}^{\text{AFHC}}(1 : T) = c(1) + c(2) + c(3) + c(4) + 2\bar{a} + w\bar{a} + \sum_{j=2}^{T/4} [c(4j - 3) + c(4j - 2)]\bar{a} + [c(4j - 1) + c(4j)]\bar{a} + w\bar{a}.

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

\[
\text{Cost}^{\text{AFHC}}(1 : T) \geq \frac{T/4 \sum_{t=1}^{T} c(t) - q^2}{2q^2} = \frac{T\bar{a} - q^2}{16\bar{a}},
\]

which can become arbitrarily large as the time horizon \(T\) 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 very large. In this section, we will introduce a novel 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 \(\tau\) be an integer from 0 to \(K - 1\). The \(\tau\)-th version of RMHC, denoted by RMHC(\(\tau\)), plays a similar role as FH(\(\tau\)), but with some crucial differences. At each time \(t = \tau + Ku, u = 0, 1, \ldots, [\frac{T}{K}]\), recall that the current input \(A(t)\) is revealed. Further RMHC(\(\tau\)) is given the inputs in the next look-ahead window \(t + 1, \ldots, t + K\). RMHC(\(\tau\))

![Figure 2: Averaging Regularized Moving Horizon Control: at each time \(t\), it averages decisions from \(K\) versions of RMHC.](image)

then calculates the solution to the following problem \(Q^{\text{RMHC}(\tau)}(t)\):

\[
\begin{align*}
\min_{\hat{X}(t + K - 1)} & \quad \sum_{s=t}^{t+K-1} [C^T(s)\hat{X}(s) + \hat{W}(s)\hat{X}(s - 1) + \hat{W}(s)\hat{X}(s - 1)]^+ \\
- &\quad [\hat{W}T - C^T(t + K)]^+ \cdot \hat{X}(t + K - 1) \
\end{align*}
\]

subject to: \(B_1\hat{X}(s) \geq B_2\hat{X}(s),\) for all time \(s \in [t, t + K - 1]\).

\(\hat{X}(s) \geq 0,\) for all time \(s \in [t, t + K - 1].\)

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

Now we discuss the 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 \(Q^{\text{FH}(\tau)}(t)\) 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 (due to the "end-of-look-ahead" problem described in Sec. 3.1) since the future is unknown.

(ii) **Regularization Term:** In Algorithm 1, at each time \(t\), there is an additional term, i.e., \([-\hat{W}T - C^T(t + K)]^+ \cdot \hat{X}(t + K - 1)\), at the end of the objective in Eq.(9). This term can be viewed as a "regularization
Algorithm 1 Averaging Regularized Moving Horizon Control (ARMHC)

Input: $\tilde{A}(1:T), K$.
Output: $\tilde{X}_{\text{ARMHC}}(1:T)$.

FOR $t = -K + 2 : T$

Step 1: $\tau \leftarrow t \bmod K$, and then $\tilde{X}(t - 1) \leftarrow \tilde{X}_{\text{ARMHC}}^r(t - 1)$.

Step 2: Given the current input $\tilde{A}(t)$ and the near-term future inputs $\tilde{A}(t + 1 : t + K)$ in the look-ahead window, if $t + K \leq T$ then

solve problem $\tilde{Q}_{\text{ARMHC}}^r(\tau)$ in Eq.(9) to get $\tilde{X}_{\text{ARMHC}}^r(t)$.

else

solve problem $\tilde{Q}_{\text{ARMHC}}^r(t)$ in Eq.(10) to get $\tilde{X}_{\text{ARMHC}}^r(t + K)$.

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

end if

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

$$
\tilde{X}_{\text{ARMHC}}(t) = \frac{1}{K} \sum_{t=0}^{K-1} \tilde{X}_{\text{ARMHC}}^r(t).
$$

END

Our key result of this section is in Theorem 3.1, where we provide 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 $T$,

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

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. As we discussed earlier, the competitive ratio of AFHC [11] could increase arbitrarily as the switching-cost coefficient increases. Similarly, the competitive ratio of the regularization method [2] 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 attain such a clear and parameter-independent competitive ratio by utilizing perfect look-ahead. Further, even if the size of the look-ahead window $K$ takes the smallest value 1, the competitive ratio of ARMHC is already 2. Thus, this result characterizes, for the first time in the literature, the high value of even a small amount of perfect look-ahead in reducing the competitive ratio.

**Remark 2.** We note that the idea of “Go-Back-1-Step” has appeared in the literature. For example, in [5], the authors propose to compute the decision for the entire look-ahead window, but only commit to the first part of the decision. However, as we can see in the proof of Theorem 3.1, this idea is insufficient to obtain good competitive ratios. The second idea of “regularization” is also critical. 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, it is truly the combination of these two ideas that works. Finally, we note that our regularization term is different from most results in the literature [2].

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

\begin{align}
\min_{\{\tilde{X}(1:T), \tilde{Y}(1:T)\}} & \left\{ \sum_{t=1}^{T} \tilde{C}^T(t)\tilde{X}(t) + \sum_{t=1}^{T} \tilde{W}^T\tilde{Y}(t) \right\} \\
\text{sub. to: } & B_1\tilde{X}(t) \geq B_2\tilde{A}(t), \text{ for all time } t \in [1, T], \\
& \tilde{Y}(t) \geq \tilde{X}(t) - \tilde{X}(t - 1), \text{ for all time } t \in [1, T], \\
& \tilde{X}(t), \tilde{Y}(t) \geq 0, \text{ for all time } t \in [1, T].
\end{align}

Then, let $\tilde{\tilde{b}}(t) = [\tilde{b}_1(t), l = 1, \ldots, L]^T \in \mathbb{R}^{L \times 1}$ and $\tilde{\tilde{b}}(t) = [\theta_0(t), n = 1, \ldots, N]^T \in \mathbb{R}^{N \times 1}$ be the dual variables for constraints of time $t = 1, \ldots, T$, in Eq.(13b) and Eq.(13c), respectively. We have the offline dual optimization problem to (13),

\begin{align}
\max_{\{\tilde{\tilde{b}}(1:T), \tilde{\tilde{b}}(1:T)\}} & \sum_{t=1}^{T} \tilde{\tilde{b}}^T(t)B_2\tilde{\tilde{b}}(t)
\end{align}
The key of proving Theorem 3.1 is to show that RMHC produces a sequence of dual variables that are feasible to the offline dual optimization problem (14). Then, since the dual problem is a maximization problem, we have $D_{\text{RMHC}}(1 : T) \leq D^\text{OPT}(1 : T)$. If we can show that the primal variables computed by RMHC satisfy $\text{Cost}_{\text{RMHC}}((1 : T) \leq \text{CR} \cdot D_{\text{RMHC}}(1 : T)$, the desired results will follow.

According to duality theorem [1, p. 225], we know that the optimal offline dual cost $D^\text{OPT}(1 : T)$ of problem (14) is equal to the optimal primal cost of problem (13) or problem (2), i.e.,

$$D^\text{OPT}(1 : T) = \text{Cost}^\text{OPT}(1 : T).$$

The key of proving Theorem 3.1 is to show that RMHC produces a sequence of dual variables that are feasible to the offline dual optimization problem (14). Then, since the dual problem is a maximization problem, we have $D_{\text{RMHC}}(1 : T) \leq D^\text{OPT}(1 : T)$. If we can show that the primal variables computed by RMHC satisfy $\text{Cost}_{\text{RMHC}}(1 : T) \leq \text{CR} \cdot D_{\text{RMHC}}(1 : T)$, the desired results will follow.

However, we have two difficulties. First, because of the additional regularization terms in the objective of RMHC, the relationship between $\text{Cost}_{\text{RMHC}}(1 : T)$ and $D_{\text{RMHC}}(1 : T)$ 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 by RMHC): First, we can get an equivalent formulation of problem (9) in a similar manner as in the objective of the offline problem (13).

$$\min \{\tilde{X}(t:t+K-1), \tilde{Y}(t:t+K-1)\} \left\{ \sum_{s=t}^{t+K-1} \left[ \tilde{C}^T(s)\tilde{X}(s) + \tilde{W}^T\tilde{Y}(s) \right] ight\}
\begin{aligned}
&\text{subject to:} & B_1 \tilde{X}(s) &\geq B_2 \tilde{A}(s), & \text{for all } s \in [t, t + K - 1], & \text{(16b)}
& & \tilde{Y}(s) &\geq \tilde{X}(s) - \tilde{X}(s - 1), & \text{for all } s \in [t, t + K - 1], & \text{(16c)}
& & \tilde{X}(s), \tilde{Y}(s) &\geq 0, & \text{for all } s \in [t, t + K - 1]. & \text{(16d)}
\end{aligned}$$

For each round of RMHC after time $t$ to $t + K - 1$, we define the primal cost $\text{Cost}_{\text{RMHC}}(t : t + K - 1)$ is independent of $t$.

$$\text{Cost}_{\text{RMHC}}(r) (t : t + K - 1) = \sum_{s=t}^{t+K-1} \left[ \tilde{C}^T(s)\tilde{X}(s) + \tilde{W}^T\tilde{Y}(s) \right].$$

Note that $\text{Cost}_{\text{RMHC}}(t : t + K - 1)$ 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 $\bar{D}_{\text{RMHC}}^{(r)} (t : t + K - 1), \bar{D}_{\text{RMHC}}^{(r)} (t : t + K - 1)$ be the optimal dual solution to the dual of the optimization problem in Eq. (16). Then, for each $r \in [0, K - 1]$, we can define the online dual cost in a similar manner as the offline dual problem (14), i.e.,

$$\bar{D}_{\text{RMHC}}^{(r)} (t : t + K - 1) \leq \sum_{s=t}^{t+K-1} \left( \tilde{R}_{\text{RMHC}}^{(r)}(s) \right)^T B_2 \tilde{A}(s).$$

However, note that (16) contains additional regularization terms in the objective function. Thus, there will be some gap between

$$\text{Cost}_{\text{RMHC}}^{(r)} (t : t + K - 1) + \bar{D}_{\text{RMHC}}^{(r)} (t : t + K - 1).$$
online dual components with 0.) However, as we discussed above, dual variables from each row of this matrix $G^{old}$ may not form a feasible solution to the offline dual optimization problem. We let $\tau_i = \lfloor \frac{t}{K} \rfloor_k + 1 + i \mod K$, for $i = 1, 2, 3, ..., K, K + 1$. With the “re-stitching” idea, we instead re-organize them into a new matrix $G^{new}$ in Eq.(22).

Note that the non-zero terms in $G^{new}$ has a one-to-one correspondence to the terms in $G^{old}$. There is a new row at the bottom of $G^{new}$ because the corresponding terms are not included in row 1 to $K$. For example, $D^{RMHC^{(0)}}(0 : K - 1), D^{RMHC^{(1)}}(K + 1 : 2K), ...$, has not been included in row 1, 2, ..., respectively. For each 0 in $G^{new}$, we assign the dual variables as follows. For the $\tau$-th row, 0 in $G^{new}$ corresponds to time slots $\tau + (K + 1)u + K, u = 1, 0, ...$

Then we use \( \bar{\beta}(\tau + (K + 1)u + K) = \bar{\beta}(\tau + (K + 1)u) = \lfloor \bar{W} - \bar{C}(\tau + (K + 1)u + K) \rfloor_k \) for each time $\tau + (K + 1)u + K$ between $\Delta^{RMHC^{(r)}}(\tau + (K + 1)u) : \tau + (K + 1)(v + K) + 1 - 1)$ and $\Delta^{RMHC^{(r)}}(\tau + (K + 1)(v + K) + 1 - 1)$, where $r = (\tau + (K + 1)v) \mod K$ and $r' = (\tau + (K + 1)(v + K) + 1) \mod K$. Note that, since in $\tau$-th row $\bar{\beta}(\tau + (K + 1)u + K) = 0$, the corresponding dual cost added to the original online dual costs is zero, which is why we simply write 0 in $G^{new}$.

Let $G_{i,j}^{old}$ be the component on the $i$-th row, $j$-th column of a matrix $G$. Then, Lemma 3.3 shows two important properties of $G^{new}$.

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

$$
\sum_{i=1}^{K} \sum_{j=1}^{K} G_{i,j}^{old} \leq \sum_{i=1}^{K} \sum_{j=1}^{K} G_{i,j}^{new}.
$$

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

Part (1) of Lemma 3.3 holds because we are only rearranging the terms of $G^{old}$ into $G^{new}$. The key of checking part (2) of Lemma 3.3 is to verify the constraint (14b) at time $t + K - 1$ and $t + K$. Indeed, we can show that $\tilde{\beta}(t+K) = [\tilde{W} - \tilde{C}(t+K+1)]^{+}$ and $\tilde{\beta}(t+K+1) = \tilde{W}$. The feasibility constraint can then be verified. Please see Appendix D for the proof of Lemma 3.3. Satisfying this constraint is also the reason why we choose the regularization term in the manner in (9). See the discussion in Appendix D.

**Step 3:** We next show in Lemma 3.4 that the sum of the tail-terms from the same version of $\Delta^{RMHC(r)}$ over all rounds could be upper-bounded by a constant 0.

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

$$
\sum_{u=1}^{\lfloor \frac{t}{K} \rfloor_k} \left[ \psi^{(r)}(\tau + Ku - 1) + \psi^{(r)}(\tau + Ku + K) \right] \leq 0.
$$

This is mainly because $[\tilde{W}^{T} - \tilde{C}^{T}(t+K)]^{+} \Delta^{RMHC^{(r)}}(t + K - 1) \leq \tilde{W}^{T} \Delta^{RMHC^{(r)}}(t - 1)$ and then most terms in the summation can be cancelled. 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.(2), we know

$$
\text{Cost}^{ARMHC}(1 : T) = \sum_{t=1}^{T} \tilde{C}^{T}(t) \tilde{X}^{ARMHC}(t)
$$

$$
\quad + \sum_{t=1}^{T} \tilde{W}^{T} \left[ X^{ARMHC}(t) - \tilde{X}^{ARMHC}(t - 1) \right]^{+}.
$$

Then, from Eq.(11) that ARMHC takes the average of $\Delta^{RMHC}$ versions of RMHC decisions, we have

$$
\text{Cost}^{ARMHC}(1 : T) = \sum_{t=1}^{T} \left\{ \tilde{C}^{T}(t) \cdot \frac{1}{K} \sum_{\tau=0}^{K-1} \Delta^{RMHC^{(r)}}(t) \right\}
$$

$$
\quad + \sum_{t=1}^{T} \left\{ \tilde{W}^{T} \left[ \frac{1}{K} \sum_{\tau=0}^{K-1} \Delta^{RMHC^{(r)}}(t) - \frac{1}{K} \sum_{\tau=0}^{K-1} \Delta^{RMHC^{(r)}(t - 1)} \right]^{+} \right\}.
$$

8
Due to the convexity of \( X^{ARMHC}(t+1) \) and Jensen’s Inequality, we have
\[
\text{Cost}^{ARMHC} (1 : T) \leq \frac{1}{K} \sum_{t=1}^{T} \left\{ \sum_{\tau=0}^{K-1} C^{t}(t) X^{ARMHC}^{\tau}(t) \right\} + \frac{\tau}{K} \sum_{t=1}^{T} \left\{ \sum_{\tau=0}^{K-1} W^{t}_{\tau} \left[ X^{ARMHC}^{\tau}(t) - X^{ARMHC}^{\tau}(t-1) \right]^{+} \right\}.
\]

By changing the order of the summations over \( t \) and \( \tau \), and put all rounds of RMHC\((\tau)\) before re-stitching together, we get
\[
\text{Cost}^{ARMHC} (1 : T) \leq \frac{1}{K} \sum_{\tau=0}^{K-1} \left\{ \sum_{u=1}^{T} \left[ \phi^{\tau}(\tau + Ku + K - 1) \right] \right\} + \phi^{\tau}(\tau + Ku - 1) + \phi^{\tau}(\tau + Ku + K - 1)\right\}.
\]

For the terms involving the online dual cost \( D^{\text{RMHC}}(\tau + Ku : \tau + Ku + K - 1) \), applying the re-stitching idea, we have,
\[
\sum_{\tau=0}^{K-1} \sum_{u=1}^{T} \left[ \phi^{\tau}(\tau + Ku + K - 1) \right] = \sum_{i=1}^{K} \sum_{j=1}^{T} C_{i,j}^{\text{old}} + \sum_{i=1}^{K} \sum_{j=1}^{T} C_{i,j}^{\text{new}} \leq (K + 1) D^{\text{OPT}} (1 : T),
\]
where the step Eq.(30a) follows from part (1) of Lemma 3.3, and the step Eq.(30b) follows from part (2) of Lemma 3.3 and the dual objective value of any feasible dual variables must be smaller than the optimal offline dual \( D^{\text{OPT}} (1 : T) \). Further, by Lemma 3.4, we know the rest of the terms in Eq.(29) satisfies
\[
\sum_{\tau=0}^{K-1} \sum_{u=1}^{T} \left[ \phi^{\tau}(\tau + Ku + K - 1) \right] \leq 0.
\]

Combining Eqs.(29), (30) and (31) and due to Eq.(15), we thus have
\[
\text{Cost}^{ARMHC} (1 : T) \leq \frac{1}{K} \cdot (K + 1) D^{\text{OPT}} (1 : T) \leq \frac{K + 1}{K} \cdot \text{Cost}^{\text{OPT}} (1 : T), \text{ for all } T \text{ and } \tilde{A}(1 : T).
\]

### 3.3 Convex Service Cost

The above results assume that the service cost is linear in \( \tilde{X}(t) \). For the case with perfect look-ahead, our results can be generalized to convex service-cost functions, with a minor change of ARMHC. Specifically, here the service cost at time \( t \) is given by \( h_{\tau}(\tilde{X}(t)) \), where \( h_{\tau}(\cdot) \) is a convex function. In Algorithm 1, instead of solving the optimization problem (9), ARMHC solves the optimization problem (33) below with convex service costs:
\[
\min_{\tilde{X}(t+K-1)} \left\{ \sum_{s=t}^{t+K-1} h_{\tau}(\tilde{X}(s)) + \tilde{W}^{T} [\tilde{X}(s) - \tilde{X}(s-1)]^{+} \right\}
\]
\[
- \left[ \tilde{W}^{T} - \nabla h_{\tau+K}[\tilde{X}(t+K) = B_{1}^{\tau} B_{2} \tilde{A}(t+K)]^{+} \tilde{X}(t + K - 1) \right]
\]
\begin{align}
& \text{sub. to:} B_{1} \tilde{X}(s) \geq B_{2} \tilde{A}(s), \text{ for all time } s \in [t, t + K - 1], \nonumber \\
& \tilde{X}(s) \geq 0, \text{ for all time } s \in [t, t + K - 1], \nonumber \\
& \tilde{X}(s) \geq 0, \text{ for all time } s \in [t, t + K - 1], \nonumber \\
& \text{where } \nabla h_{t+K} = \left[ \frac{\partial h_{t+K}}{\partial x_{n}(t+K)}, n = 1, ..., N \right]^{T} \in \mathbb{R}^{N \times 1} \text{ and } B_{1}^{\tau} \text{ is the pseudo-inverse of the matrix } B_{1}. \nonumber \\
& \text{Note that the regularization term is changed to } \nonumber \\
& - \left[ \tilde{W}^{T} - \nabla h_{t+K}[\tilde{X}(t+K) = B_{1}^{\tau} B_{2} \tilde{A}(t+K)]^{+} \tilde{X}(t + K - 1) \right].
\end{align}

The choice of \( \tilde{X}(t + K) = B_{1}^{\tau} B_{2} \tilde{A}(t + K) \) helps to ensure that the hard constraint at time \( t + K \) is satisfied.

Similarly, RMHC replaces the service cost by \( h_{\tau}(\tilde{X}(t)) \) for Eq.(10) when \( t + K > T \). The performance of ARMHC is given in Theorem 3.5.

**Theorem 3.5.** For the online convex optimization problem formulated in Sec. 2.1 with convex service costs \( h_{\tau}(\tilde{X}(t)) \), the Averaging Regularized Moving Horizon Control (ARMHC) algorithm described above is \((\frac{K+1}{K})\)-competitive.

Please see Appendix H for the proof of Theorem 3.5. The main difficulty of proving Theorem 3.5 is to verify Lemma 3.2 with convex service costs (See discussion and the proof in Appendix G) and check the feasibility of the new group of duals after applying the re-stitching idea, i.e., Lemma 3.3. In Appendix F, we provide a sufficient condition the feasibility of dual variables.

### 4 THE IMPERFECT LOOK-AHEAD CASE

In the previous section, we have assumed that the future information in the look-ahead window of \( K \) slots 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. ??, where the decision maker is given the current input along with a predicted input trajectory \( \bar{A}_{pred}^{t+1}(t + 1 : t + K) \) in the look-ahead window. Moreover, she knows the upper and lower bounds defined by \( \bar{A}_{up}^{t+1}(t + 1 : K) \) and \( \bar{A}_{low}^{t+1}(t + 1 : K) \) as in Eq.(3) and (4). These bounds \( \bar{A}_{up}^{t+1}(t + 1 : K) \) and \( \bar{A}_{low}^{t+1}(t + 1 : K) \) provide us with the information on how far the real input in the look-ahead window could deviate from the predicted input trajectory. 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) algorithm in [5] also aims to deal with imperfect look-ahead. However, the results there only provide guarantees on the competitive difference of CHC, not its competitive ratios. Further, the results there assume no hard constraints. Thus, it remains an open problem to develop online algorithms with low competitive ratios under imperfect look-ahead and hard constraints.

4.1 Dealing with Hard Constraints under Imperfect Look-Ahead

In the perfect look-ahead case, the predicted inputs in the look-ahead window is precise. Hence, the decisions solved by RMHC(r) in Eq.(9) always satisfy the real inputs, so do the ARMHC decisions when an average is taken over different versions of RMHC(r). 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. Then, if we still use the predicted input in the constraints in Eq.(9), the decisions obtained may not satisfy the hard constraints with real inputs. Thus, ARMHC may fail to satisfy hard constraints either.

A straightforward way to fix this difficulty is to use the upper bound of the future inputs in the optimization problem (9). Thus, at each time $t$, RMHC(r) should consider the constraints with the upper bound of the future inputs, i.e., replace (9b) and (10b) by

$$B_1\bar{X}(s) \geq B_2\bar{A}(s), \text{ for all time } s \in [t, t+K-1],$$

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

**Theorem 4.1.** The competitive ratio of Averaging Regularized Moving Horizon Control-Imperfect Look-ahead (ARMHC-IL) algorithm is

$$\frac{K+1}{K} \max_{\{i \in [0, K-1], m \in [1, M]\}} \left\{ \frac{\rho_i(m)}{\rho_i(m+1)} \right\}.$$  

To the best of our knowledge, ARMHC-IL algorithm is the first to achieve a theoretically-guaranteed competitive ratio under imperfect look-ahead for online convex optimization (OCO) problem with 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. 3.2. The second part is based on the gap between the upper and lower bounds. Thus, it captures the impact of prediction errors. If there are no prediction errors, the second part is 1. We then recover the results with perfect look-ahead. Please see Appendix I for the proof of Theorem 4.1.

However, the competitive ratio of ARMHC-IL is still highly sensitive to the bounds on the prediction accuracy. Consider the following case where, within the look-ahead window of size $K$, some time slots have better prediction accuracy, i.e., the value of $\frac{\rho_i(m)}{\rho_i(m+1)}$ is small, but other time-slots have much poorer prediction accuracy.

Then, according to Theorem 4.1, the competitive ratio of ARMHC-IL will be dominated by the worst quality of prediction.

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(r) differently, according to how reliable the prediction is. The following Weighting Regularized Moving Horizon Control (WRMHC) algorithm assigns different weights to inputs and decisions, which can achieve a lower competitive ratio than ARMHC-IL and reduce the 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) algorithm 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 qualities of predictions 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 quality of predictions at the end of the look-ahead window is very poor. Specifically, let the size of the committed look-ahead window be $k$, $1 \leq k \leq K$. 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 $r$ be an integer from the set $[0, K-1]$. At each time $t = r + ku, u = -1, 0, ..., \frac{T}{K}$, the $\tau$-th version of GRMHC, denoted by GRMHC(r), is given the current input $\tilde{A}(t)$ and the predicted inputs $\tilde{A}^{\text{pred}}_i(t+1 : t+k)$. Then, GRMHC(r) calculates the solution to the following optimization problem $Q^{\text{GRMHC(r)}}(t)$:

$$\min_{\tilde{X}(t+k-1)} \left\{ \sum_{s=t}^{t+k-1} \rho(s-t)\tilde{C}_t(s)\tilde{X}(s) + \tilde{W}_T \begin{bmatrix} \rho(0)\tilde{X}(t) - \rho(k-1)\tilde{X}(t-1) \end{bmatrix}^+ \right. + \sum_{s=t+1}^{t+k-1} \tilde{W}_T \begin{bmatrix} \rho(s-t)\tilde{X}(s) - \rho(s-t-1)\tilde{X}(s-1) \end{bmatrix}^+ \\
- \begin{bmatrix} \tilde{W}_T - \tilde{C}_T(t+k) \end{bmatrix}^+ \cdot \rho(k-1)\tilde{X}(t+k-1) \right\}$$

subject to:

$$B_1\tilde{X}(t) \geq B_2\tilde{A}(t),$$

$$B_1\tilde{X}(s) \geq B_2\tilde{A}(s), \text{ for all time } s \in [t+1, t+k+1],$$

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

Note that a weight $\rho(i)$ is assigned to each time-slot in the objective of (36). Intuitively, with a smaller $\rho(i)$, the cost of the corresponding time 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(r) only commits to the first $k$-slots of the entire available look-ahead window, the initial value $\tilde{X}(t-1)$ for $Q^{\text{GRMHC(r)}}(t)$
is from GRMHCr(t) computed at an earlier time of \( t - k \), where \( QGMHC^{cr}(t - k) \) was solved. Let \( \tilde{X}(t, i) \triangleq (t - i) \text{mod} k \) for any \( i \in [0, k - 1] \). Then, at each time \( t \), WRMHC takes the weighted average of \( \widehat{X}GMHC^{cr}(t) \) for all \( r \in [0, k - 1] 
\[
\widehat{X}WRMHC(t) = \sum_{i=0}^{k-1} \rho(i) \widehat{X}GMHC^{cr}(t,i)(t) \quad \text{for all} \quad r \in [0, k - 1].
\]
The details are given in Algorithm 2.

**Algorithm 2 Weighting Regularized Moving Horizon Control (WRMHC) with committed look-ahead window of size \( k \)**

**Input:** \( \bar{A}(1 : T), K, \hat{f}^{up}(1 : K) \) and \( \hat{f}^{low}(1 : K) \).
**Output:** \( \widehat{X}WRMHC(1 : T) \).

FOR \( t = -k + 2 : T \)

Step 1: \( \tau \leftarrow \text{mod} k \), and then \( \tilde{X}(t - 1) \leftarrow \tilde{X}GMHC^{cr}(t - 1) \).

Step 2: Given the current input \( \bar{A}(t) \) and predicted near-term future inputs \( \bar{A}_{pred}(t + 1 : t + K) \) in the committed look-ahead window,

if \( t + k \leq T \) then

solve problem \( \tilde{Q}GMHC^{cr}(t) \) in Eq.(36) to get \( \tilde{X}GMHC^{cr}(t) \),

else

solve problem \( \tilde{Q}GMHC^{cr}(t) \) in Eq.(38) to get \( \tilde{X}GMHC^{cr}(t) \),

\[
\min_{\tilde{X}(t:T)} \left\{ \sum_{s=t}^{T} \rho(s-t) \hat{C}^T(s)\tilde{X}(s) + \tilde{W}^T \rho(0)\tilde{X}(t) - \rho(k-1)\tilde{X}(t-1) \right\}^+
+ \sum_{s=t+1}^{T+k-1} \tilde{W}^T \rho(s-t)\tilde{X}(s) - \rho(s-t)\tilde{X}(s-1) \right\}^+
\]

sub. to: \( B_{1}\tilde{X}(s) \geq B_{2}\tilde{A}(s) \), for all time \( s \in [t + 1, T] \),
\( \tilde{X}(s) \geq 0 \), for all time \( s \in [t, T] \).

end if

Step 3: Use Eq.(37) to get the final decision \( \tilde{X}WRMHC(t) \).

END

Similar to RMHC, if \( t + K > T \), GRMHCr solves \( \tilde{Q}GMHC^{cr}(t) \) in Eq.(38) instead of \( QGMHC^{cr}(t) \) in Eq.(36).

In summary, comparing WRMHC with ARMHC-IL, we can see two important differences. The first one is the different weights assigned to terms of each service costs and switching costs in the objective of (36). These weights represent different importance of cost-terms due to different qualities of predictions. The second one is the different weights in the averaging in (37), which again reflect different importance due to different qualities of prediction. Theorem 4.2 below gives the competitive ratio of WRMHC.

**Theorem 4.2.** Given the bounds \( \hat{f}^{up}(1 : K) \) and \( \hat{f}^{low}(1 : K) \), the total cost of the Weighting Regularized Moving Horizon Control (WRMHC) algorithm with the committed look-ahead window of size \( k \) and weights \( \rho(0 : k - 1) \) satisfies, for any \( T \),

\[
\text{Cost}_{WRMHC}(1 : T) \leq \frac{k + 1}{k - 1} \cdot \max_{\{i\in[0,k-1],m\in[1,M]\}} \sum_{i=0}^{k-1} \rho(i) \cdot \text{Cost}_{OPT}(1 : T), \quad \text{for all} \quad A(1 : T).
\]

Theorem 4.2 provides the competitive ratio of the WRMHC algorithm. This competitive ratio can also be viewed as containing two parts. The first part, \( \frac{k+1}{k-1} \), is still similar to CRARMHC 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, \( \max_{\{i\in[0,k-1],m\in[1,M]\}} \sum_{i=0}^{k-1} \rho(i) \), is not only based on the gap between the upper and lower bounds, but also with the weights. Thus, it captures the combined impact of both prediction errors and the weights used in WRMHC. This result then allows us to set appropriate weights to offset the influence from poor predictions. Please see Appendix J 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 the competitive ratio in Eq.(39) is

\[
\rho^*(i) = \min_{m\in[1,M]} \frac{f^{up}_{m}(i)}{f^{low}_{m}(i)} \quad \text{for all} \quad i \in [1, k - 1].
\]

This is true by verifying these weights

\[
\rho^*(0 : k - 1) = \arg \min_{\rho(0 : k - 1) \geq 0} \left\{ \frac{k + 1}{k - 1} \cdot \max_{\{i\in[0,k-1],m\in[1,M]\}} \sum_{i=0}^{k-1} \rho(i) \cdot \text{Cost}_{OPT}(1 : T) \right\}.
\]

See Appendix K for the proof of Lemma 4.3. Then, the competitive ratio of WRMHC becomes \( \frac{k + 1}{k - 1} \cdot \max_{\{i\in[0,k-1],m\in[1,M]\}} \sum_{i=0}^{k-1} \rho(i) \). Finally, we can obtain the optimal size of the committed look-ahead window as

\[
k^* = \arg \min_{1 \leq k \leq K} \left\{ \frac{k + 1}{k - 1} \cdot \max_{\{i\in[0,k-1],m\in[1,M]\}} \sum_{i=0}^{k-1} \rho(i) \right\}.
\]

**Theorem 4.4.** The competitive ratio of WRMHC with the weights in Eq.(40) and optimal size of committed look-ahead window \( k^* \) given in Eq.(41) is

\[
\text{CR}_{WRMHC}^* = \frac{k^* + 1}{1 + \sum_{i=1}^{k^*} \min_{m\in[1,M]} \frac{f^{up}_{m}(i)}{f^{low}_{m}(i)}}.
\]

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 \( \max_{\{i\in[0,k-1],m\in[1,M]\}} \left\{ \frac{f^{up}_{m}(i)}{f^{low}_{m}(i)} \right\} \). For example, the competitive ratio in Eq.(42) will never be larger than \( k^* + 1 \), while the competitive ratio in Theorem 4.1 can potentially be much higher.
larger than $k^* + 1$. This insensitivity is because WRMHC assigns a much smaller weight $\rho(i) = \frac{f_{m}^{up}(i)}{f_{c}^{up}(i)}$ for the worst-case prediction.

Further, from Algorithm 2 we notice that when the look-ahead is perfect, i.e., $f_{m}^{up}(i) \equiv 1$ and $f_{m}^{low}(i) \equiv 1$ 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 $k^* = K$. 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 EVALUATION

In this section, we compare the theoretical competitive ratios of our proposed algorithms and the existing algorithms under perfect and imperfect look-ahead. Using traces either from Counter-example 1 or from the HP trace at [7], we also provide numerical results for comparing the empirical performance of ARMHC and AFHC for perfect look-ahead, and WRMHC and CHC for imperfect look-ahead to show the superior performance of ARMHC and WRMHC for different inputs. See Appendix L for more simulation results.

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 [11] for perfect look-ahead and its competitive ratio is shown to be $1 + \max_{n \in [1, N], t \in [1, T]} \frac{W_n}{C_n(t)}$. In contrast, the competitive ratio of our proposed ARMHC is $\frac{K+1}{K}$. These competitive ratios are plotted in Fig. 3b and Fig. 3a, respectively. In Fig. 3b, we simply use the label $w$ to represent $\max_{n \in [1, N], t \in [1, T]} \frac{W_n}{C_n(t)}$.

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 $\frac{w}{c}$ is very large. For example, when $K = 2$, $CR_{AFHC} = 7.667$ and $17.667$ for $w/c = 20$ 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) problem with perfect look-ahead.

Then, we provide numerical results. We take $T = 336$. The service-cost coefficient $\hat{C}(t)$ and the switching-cost coefficient $\hat{W}$ are generated randomly in advanced. While $\hat{W}$ is fixed, we allow $\hat{C}(t)$ to change in time. $\hat{C}(t)$ are uniformly generated in the continuous interval from 0 to 1, i.e., $\hat{C}(t) \sim U(0, 1)$. For Fig. 4a and 5b, $\hat{W} \sim U(3, 4)$, thus, $\hat{W} \geq 3\hat{C}(t)$. For Fig. 4b and 5c, $\hat{W} \sim U(20, 30)$, thus, $\hat{W} \geq 20\hat{C}(t)$. For Fig. 4c and 5d, $\hat{W} \sim U(50, 60)$, thus, $\hat{W} \geq 50\hat{C}(t)$.

In Fig. 4, we compare the empirical performance of ARMHC with that of AFHC using the inputs presented in Counter-example 1 in Sec. 3.1. Here, we take $N = 10, M = 30, L = 5$ and all entries of $B_1$ and $B_2$ 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. 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 AFHC using the inputs generated from the HP trace at [7] (re-produced in Fig. 5a). Notice that here $M = 1$. Here, we take $N = 3, L = 1$ and all entries of $B_1$ and $B_2$ are 1. We consider one week and the inputs are for every half an hour. From Fig. 5, we can observe that, with pretty small loss only when both $K$ and $\hat{W}$ are 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 ratio of ARMHC-IL and WRMHC. Here, the bounds $f_{m}^{up}(i) = 1 + \epsilon_{up}, i$ and $f_{m}^{low}(i) = \max\{0, 1 - \epsilon_{low}, t\}$, for all $i \in [1, K]$, where $\epsilon_{up} = 0.3$ and $\epsilon_{low} = 0.025$. First, we can observe that ARMHC-IL is highly sensitive to the prediction error since when $f_{m}^{up}(i)$ increases as $k$ increases, $CR_{ARMHC-IL}$ increases quickly. In contrast, the competitive ratio of WRMHC increase much slower and keep around a value of 2. Second, ARMHC-IL can also choose the optimal size of a committed look-ahead window, i.e., $CR_{ARMHC-IL} = 2$ when choosing $k^* = 2$. However, we can still get a better competitive ratio from WRMHC* that $CR_{WMHC*} = 1.7$. 

![Figure 3: Theoretical competitive ratio (CR) of AFHC and ARMHC.](image1)

![Figure 6: Empirical competitive ratio (ECR) of AFHC and ARMHC for inputs in Counter-example 1.](image2)
In this paper, we study how to best utilize either perfect or imperfect look-ahead to achieve low competitive ratio in online convex optimization (OCO) problem. For perfect look-ahead, we develop the Averaging Regularized Moving Horizon Control (ARMHC) algorithm, which is $\frac{1}{1+\epsilon}$-competitive. This is the first result in the literature that can achieve parameter-independent competitive ratio using perfect look-ahead. For imperfect look-ahead, we provide the Averaging Regularized Moving Horizon Control-Imperfect Look-ahead (ARMHC-IL) algorithm. We propose Weighting Regularized Moving Horizon Control (WRMHC) algorithm, whose competitive ratio is less sensitive to the prediction errors. It is the first algorithm for geographical load balancing. In Green Computing Conference (IGCC), 2012 International, IEEE, 1–10.

In the following, we provide the derivation of decision sequences for ARMHC-IL in Counter-Example 1. In Algorithm 3, we provide the details of Averaging Fixed Horizon Control (AFHC) algorithm.

In the following, we provide the derivation of decision sequences of different versions of FHC.

**REFERENCES**


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

In Algorithm 3, we provide the details of Averaging Fixed Horizon Control (AFHC) algorithm.

In the following, we provide the derivation of decision sequences of different versions of FHC.
Algorithm 3 Averaging Fixed Horizon Control (AFHC)

Input: \( \hat{A}(1 : T), K \).
Output: \( \overline{X}_{AFHC}^{K}(1 : T) \).
FOR \( t = -K + 1 : T \)
\( \text{Step 1: } \tau \leftarrow t \bmod (K + 1) \), then \( \overline{X}(t-1) \leftarrow \overline{X}_{AFHC}^{K}(t-1) \).
\( \text{Step 2: } \) Based on \( \hat{A}(t : t + K) \), solve problem \( Q^{AFHC}^{K}(t) \) in Eq.(7) to get \( \overline{X}_{AFHC}^{K}(t : t + K) \).
\( \text{Step 3: } \) Use Eq.(43) below to compute the final decision at time \( t \),
\[
\overline{X}_{AFHC}^{K}(t) = \frac{1}{K + 1} \sum_{r=0}^{K} \overline{X}_{AFHC}^{K}(t). \quad (43)
\]
END

AFHC(2) sees the input sequence \( (0, 0, 0, 0) \) in the first round and gives the decision \( (0, 0, 0, 0) \) to satisfy the input \( \overline{a} \) at time \( t = 1 \). She sees the input sequence \( (\overline{a}, 0, 0, \overline{a}) \) in all subsequent rounds and gives the decision sequence \( (\overline{a}, 0, 0, \overline{a}) \) for these rounds.

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

FHC(1) sees the input sequence \( (0, 0, 0, 0) \) in all subsequent rounds and gives the decision sequence \( (0, 0, 0, 0) \) for these rounds. Thus, FHC(1) keeps the initial value of the decision \( \overline{a} \), the decisions of FHC(0) and FHC(1) go down in each round.

Algorithm 4 Averaging Fixed Horizon Control-first \( K \) decisions committed (AFHC-1)

Input: \( \hat{A}(1 : T), K \)
Output: \( \overline{X}_{AFHC-1}^{K}(1 : T) \)
FOR \( t = -K + 2 : T \)
\( \text{Step 1: } \tau \leftarrow t \bmod K \), then \( \overline{X}(t-1) \leftarrow \overline{X}_{AFHC-1}^{K}(t-1) \).
\( \text{Step 2: } \) Based on \( \hat{A}(t : t + K) \), solve problem \( Q^{AFHC-1}^{K}(t) \) in Eq.(7) and commits to the first \( K \) decisions \( \overline{X}_{AFHC-1}^{K}(t : t + K - 1) \).
\( \text{Step 3: } \) Use Eq.(44) below to get the final decision
\[
\overline{X}_{AFHC-1}^{K}(t) = \frac{1}{K} \sum_{r=0}^{K-1} \overline{X}_{AFHC-1}^{K}(t). \quad (44)
\]
END

Figure 7: Behaviors of AFHC-1 with the same inputs as in Counter-example 1.

B COUNTER-EXAMPLE 2
Note that the "Go-Back-1-Step" idea is somewhat similar to the idea of CHC (Committed Horizon Control), where the decisions for the entire look-ahead window is computed, but only the first part of the decisions is committed. While we could apply the 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 AFHC-1 and FHC-1 algorithm. Specifically, at each time \( t = t + K \), \( u = -1 \), \( 0, ..., \left( \frac{\overline{X}(t)}{X} \right) \), FHC-1 is given the current input and the inputs in the current look-ahead window \( t + 1, ..., t + K \). FHC-1 then calculates the solution to the optimization problem \( Q^{FHC-1}^{K}(t) \) in Eq(7), but she only commits to the first \( K \) decisions. Thus, similar to RMHC, now the initial value \( \overline{X}(t - 1) \) for \( Q^{FHC-1}^{K}(t) \) is from FHC-1 computed at an earlier time of \( t - K \), i.e., when \( Q^{FHC-1}^{K}(t - K) \) 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 also have an arbitrarily large competitive ratio.

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 FHC-1, FHC(1), FHC(2), are plotted in Fig. 7b for \( \overline{a} = 1000 \) and \( g = 0 \). Specifically, FHC(1) sees input sequence \( (\overline{a}, 0, 0, 0) \) at time \( t = 1 \). Similar to FHC(1), she gives the...
decision sequence \((\tilde{a}, \tilde{a}, \tilde{a}, \tilde{a})\) since there is no need to maintain a large decision at time \(t = 3\) and \(4\). However, based on the “Go-
Back-1-Step” idea, she only commits to the first 3 slots, i.e., \((\tilde{a}, \tilde{a}, \tilde{a})\). Then, at time \(t = 4\), she sees input \(a(4) = \tilde{a}\). Since \(x(3) = a\), she
gives the decision \(x(4) = a\). In this way, a lower decision \(a\) is still
used at time \(t = 3\) and \(4\). The same behavior for \(\text{FHC}_{-1}^{(3)}\) at time \(t = 11, 12\) and \(\text{FHC}_{-1}^{(2)}\) at time \(t = 7, 8\). When \(\text{AFHC}_{-1}\)
takes the average of these 3 versions of \(\text{FHC}_{-1}\), the final decision sequence
becomes \((\tilde{a}, \tilde{a}, \tilde{a}, \tilde{a}, \tilde{a}, \tilde{a}, \tilde{a}, \tilde{a}, \tilde{a}, \tilde{a}, \tilde{a})\) for every round of 4 slots starting
from time \(t = 1\). We plot the decision sequence of the final \(\text{AFHC}_{-1}\)
algorithm in Fig. 7a. The corresponding total cost is \(\text{Cost}^{\text{AFHC}_{-1}} =
\left[\left(1 + c(2)\right)\tilde{a} + \left(3 + c(4)\right)\hat{a} + w/2 + \sum_{j=2}^{T/4}\left(\left(4j - 3\right) + c(4j - 2)\right)\tilde{a} + \left(4j - 1\right) + c(4j)\right]\frac{2\hat{a} + a}{\frac{1}{2}} + w\hat{a} + \sum_{j=2}^{T/4}\left(\left(4j - 3\right) + c(4j - 2)\right)\tilde{a} + \left(4j - 1\right) + c(4j)\right]\frac{2\hat{a} + a}{\frac{1}{2}} + w\hat{a}.\) Notice that the total switching cost
of \(\text{AFHC}_{-1}\) still increases by a value of \(w\hat{a}/2\) every 4 time-slots
due to the terms \(\sum_{j=2}^{T/4}\left(\left(4j - 3\right) + c(4j - 2)\right)\tilde{a} + \left(4j - 1\right) + c(4j)\right]\frac{2\hat{a} + a}{\frac{1}{2}} + w\hat{a}\). Thus, similar to Counter-example 1, if \(w \geq \sum_{t=1}^{T} c(t)\), the ratio between the total cost of \(\text{AFHC}_{-1}\) and that of the optimal solution
\[
\text{Cost}_{\text{AFHC}_{-1}}^{\text{Opt}}(1 : T) = \frac{T}{2}\frac{\tilde{a} - \hat{a}}{\frac{1}{2}} = \frac{T}{24}\frac{\tilde{a} - \hat{a}}{\frac{1}{2}}.
\]
which can become arbitrarily large as the time horizon \(T\) increases.

C \ PROOF OF LEMMA 3.2

Proof. According to Karush-Kuhn-Tucker (KKT) conditions of the dual
optimization problem of Eq.(16), we have the following equations
(for complementary slackness) that for all time \(t \in [K + 1]\),
\[
(b_{R}^{\text{RMHC}(t)}(s))^{T}\left[B_{2}(s) - B_{1}^{\text{X}^{\text{RMHC}}(s)}(s)\right] = 0, \quad (46a)
\]
\[
(b_{R}^{\text{RMHC}(t)}(s))^{T}\left[X^{\text{RMHC}}(s) - X^{\text{RMHC}}(s)\right] = 0, \quad (46b)
\]
and the following equations (for stationarity/optimality),
\[
(X^{\text{RMHC}}(s))^{T}\left[c_{s}(s) - B_{1}^{\text{Y}^{\text{RMHC}}(s)}(s) + \theta^{\text{RMHC}}(s)\right] = 0, \quad (47a)
\]
\[
(X^{\text{RMHC}}(t + K - 1))^{T}\left[c_{s}(s) - B_{1}^{\text{Y}^{\text{RMHC}}(s)}(s) + \theta^{\text{RMHC}}(s)\right] = 0, \quad (47b)
\]
\[
(Y^{\text{RMHC}}(s))^{T}\left[W - Y^{\text{RMHC}}(s)\right] = 0, \quad \forall s \in [K, t + K - 1]. \quad (47c)
\]

We know the real cost of each version of \(\text{RMHC}(t)\) in the current
decision round from any time \(t\) to \(t + K - 1\), where \(t = \tau + Ku\) and
\[
u = -1, 0, \ldots, \left\lfloor \frac{T}{K} \right\rfloor.
\]
\[
\begin{align*}
\text{Cost}_{\text{RMHC}}^{\text{RMHC}(t)}(t : t + K - 1) & = \sum_{s=t}^{t+K-1} \left[c_{s}(s)Y^{\text{RMHC}(s)}(s) + W_{s}^{T}\left[X^{\text{RMHC}(s)}(s) - X^{\text{RMHC}(s)}(s)\right]^{T}\right) \quad (48) \\
& = \sum_{s=t}^{t+K-1} \left[c_{s}(s)Y^{\text{RMHC}(s)}(s) + W_{s}^{T}\left[X^{\text{RMHC}(s)}(s) - X^{\text{RMHC}(s)}(s)\right]^{T}\right).
\end{align*}
\]
Then, according to Eqs.(46a) and (46b), by adding these zero-value
terms, add and deduct a same term \(1W - \bar{c}(t + K))^{T}X^{\text{RMHC}(s)}(t + K - 1)\), we have \(\text{Cost}_{\text{RMHC}}^{\text{RMHC}(t)}(t : t + K - 1)\)
\[
= \sum_{s=t}^{t+K-1} \left[c_{s}(s)Y^{\text{RMHC}(s)}(s) + W_{s}^{T}\left[X^{\text{RMHC}(s)}(s) - X^{\text{RMHC}(s)}(s)\right]^{T}\right)
\]
\[
- [\bar{W} - \bar{c}(t + K)]^{T}X^{\text{RMHC}(s)}(t + K - 1)
\]
\[
+ [\bar{W} - \bar{c}(t + K)]^{T}X^{\text{RMHC}(s)}(t + K - 1)
\]
\[
+ \sum_{s=t}^{t+K-1} \left[c_{s}(s)Y^{\text{RMHC}(s)}(s) + W_{s}^{T}\left[X^{\text{RMHC}(s)}(s) - X^{\text{RMHC}(s)}(s)\right]^{T}\right).
\]
Rearrange Eq.(49), we get \(\text{Cost}_{\text{RMHC}}^{\text{RMHC}(t)}(t : t + K - 1)\)
\[
\leq \sum_{s=t}^{t+K-1} \left(b_{R}^{\text{RMHC}(s)}(s))^{T}\left[B_{2}(s) - B_{1}^{\text{X}^{\text{RMHC}}(s)}(s)\right] - [\bar{W} - \bar{c}(t + K)]^{T}X^{\text{RMHC}(s)}(t + K - 1)
\]
\[
+ \sum_{s=t}^{t+K-2} \left(X^{\text{RMHC}}(s))^{T}\left[c_{s}(s) - B_{1}^{\text{Y}^{\text{RMHC}}(s)}(s) + \theta^{\text{RMHC}}(s)\right] + \left[b_{R}^{\text{RMHC}(s)}(s))^{T}\left[B_{2}(s) - B_{1}^{\text{X}^{\text{RMHC}}(s)}(s)\right] - [\bar{W} - \bar{c}(t + K)]^{T}X^{\text{RMHC}(s)}(t + K - 1)
\]
\[
+ \sum_{s=t}^{t+K-1} \left(Y^{\text{RMHC}}(s))^{T}\left[W - Y^{\text{RMHC}}(s)\right].
\]
According to Eqs.(47a-47c) and using the notations in Eqs.(18) and
(19a), (19b), we have \(\text{Cost}_{\text{RMHC}}^{\text{RMHC}(t)}(t : t + K - 1)\)
\[
\leq D_{\text{RMHC}}^{\text{RMHC}(t)}(t : t + K - 1) - \bar{W}(t + 1) + \psi(t + 1.5).
\]

D \ PROOF OF LEMMA 3.3

Proof. For (1): there are \(K\) versions of \(\text{RMHC}\), so there
are \(K\) rows in the original group \(O^{\text{Opt}}\). The total time horizon is at
most from \(t = -K + 2\) to \(t = T\) and each \(\text{RMHC}(t)\) proceeds every
\(K\) time slots, so there are at most \(\left\lfloor \frac{T}{K} \right\rfloor + 2\) components in each row.
Thus, the sum on the left hand side in Eq.(23) represents the sum of
all the components in the original group $G_{old}$.

Second, since there are some components from $G_{old}$ that is not
considered after re-stitching and we concatenate them in $(K + 1)$-th
row, there are $K + 1$ rows in the new group $G_{new}$. Between every
two adjacent components, we add a zero-value component due to
the dual variables added during re-stitching, so there are at most
\[ \left\lceil \frac{T^2}{K}\right\rceil + 2 \cdot 2 \cdot \left\lfloor \frac{T^2}{K}\right\rfloor + 3 \text{ components in each row of the}
new group $G_{new}$. Thus, the sum on the right hand side in Eq.(23)
represents the sum of all the components in the new group $G_{new}$.

Finally, during re-stitching, we only add non-negative components
does 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.(23) is true.

For (2): notice that in each row of $G_{new}$, an online dual and a zero
term appears repeatedly. Without loss of generality, considering
the online dual costs $D_{RMHC}^{(t)}(t + K - 1)$, $D_{RMHC}^{(t+1)\mod K} (t + K + 1 :
t + 2K)$, where $t = \tau + Ku$, $u = 0, 1, \ldots, \left\lfloor \frac{T}{K}\right\rfloor$. We focus on proving
the dual variables from time $t$ to time $t + K$ are feasible to the offline
dual optimization problem (14). For all the other time slots, we can
consider them in every round of $K + 1$ slots, e.g., from time $t + K - 1
to t - 1$ and from time $t + K + 1$ to $t + 2K + 1$, and the proof is the same.

(i) For from time $t$ to time $t + K - 2$, the dual variables are feasible
since they are directly from solving the dual optimization problem
subject to the same constraints (14b-14d) from time $t$ to $t + K - 2$.

For $t + K - 1$ and $t + K$, we only need to show constraint (14b)
holds since constraints (14c) and (14d) are obviously true for the
online dual variables.

(ii) For time $t + K - 1$, consider the dual variables at time $t + K - 1$
and $t + K$, i.e., $\beta_{RMHC}^{(t + K - 1)}$, $\beta_{RMHC}^{(t + K)}$ and $\tilde{\theta}(t + K)$.
We have

\[
\tilde{C}(t + K) = \tilde{C}(t + K - 1) + B_1^T b_{RMHC}^{(t + K - 1)} + \tilde{\theta}(t + K + 1) - \tilde{\theta}(t + K)
\]

\[
= \tilde{C}(t + K - 1) + B_1^T b_{RMHC}^{(t + K - 1)} + \tilde{\theta}(t + K) - \tilde{\theta}(t + K + 1) + \tilde{\theta}(t + K + 1)
\]

\[
= \tilde{C}(t + K - 1) + B_1^T b_{RMHC}^{(t + K - 1)} + \tilde{\theta}(t + K) - \tilde{\theta}(t + K + 1) + \tilde{\theta}(t + K + 1)
\]

\[
\geq \tilde{C}(t + K - 1) - B_1^T b_{RMHC}^{(t + K - 1)} + \tilde{\theta}(t + K) - \tilde{\theta}(t + K + 1) + \tilde{\theta}(t + K + 1)
\]

\[
\geq \tilde{C}(t + K - 1) = \tilde{C}(t + K) + \tilde{\theta}(t + K) - \tilde{\theta}(t + K + 1) + \tilde{\theta}(t + K + 1)
\]

\[
= \tilde{C}(t + K) + \tilde{\theta}(t + K) - \tilde{\theta}(t + K + 1) + \tilde{\theta}(t + K + 1)
\]

Hence, dual variables from any row of $G_{new}$ are feasible to the
offline dual optimization problem in Eq.(14).

Remark 3: Satisfying the constraint $\tilde{\theta}(t + 1) - \tilde{\theta}(t) \leq \tilde{C}(t) - B_1^T b_{RMHC}^{(t)}$ is one of the important reasons for choosing the regularization term to be $-\tilde{W} \tilde{C} - \tilde{C}^T (t + K)\tilde{X}(t + K - 1)$. Specifically, first, to satisfy this constraint at time $t + K$, the derivative of the regularization term needs to be

\[
\geq \beta_{RMHC}^{(t + 1)\mod K} (t + K + 1) - \tilde{C}(t + K) + B_1^T b_{RMHC}^{(t + K)}
\]

\[
\geq \beta_{RMHC}^{(t + 1)\mod K} (t + K + 1) - \tilde{C}(t + K) + B_1^T b_{RMHC}^{(t + K)} = \tilde{W} - \tilde{C}(t + K)
\]

Second, to satisfy this constraint at time $t + K - 1$, we need to verify that

\[
\tilde{C}(t + K - 1) - B_1^T b_{RMHC}^{(t + K - 1)} + \tilde{\theta}(t + K) - \tilde{\theta}(t + K + 1) + \tilde{\theta}(t + K + 1)
\]

\[
\geq \beta_{RMHC}^{(t + K - 1)} (t + K - 1) - \tilde{C}(t + K) + B_1^T b_{RMHC}^{(t + K)}
\]

Thus, the derivative of the regularization term needs to be

\[
\leq \tilde{\theta}(t + K) = \left[ \tilde{W} - \tilde{C}(t + K) \right]^T
\]

Third, we need the regularization term $\geq 0$.

Hence, we get the resulting formulation of the regularization
term $-\tilde{W} \tilde{C} - \tilde{C}^T (t + K)\tilde{X}(t + K - 1)$.

E PROOF OF LEMMA 3.4

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

\[
\sum_{u=1}^{T} \left\{ \phi^T (\tau + Ku - 1) + \psi^T (\tau + Ku + K - 1) \right\}
\]

\[
= \sum_{u=1}^{T} \left\{ -\tilde{W} \tilde{C}^{RMHC} (\tau + Ku - 1)
\right\} + \tilde{C}(t + K) + B_1^T b_{RMHC}^{(t + K)}
\]

\[
= \sum_{u=0}^{T} \left\{ -\tilde{W} \tilde{C}^{RMHC} (\tau + Ku - 1)
\right\} + \tilde{C}(t + K) + B_1^T b_{RMHC}^{(t + K)}
\]

\[
= \sum_{u=0}^{T} \left\{ -\tilde{W} \tilde{C}^{RMHC} (\tau + Ku - 1)
\right\} + \tilde{C}(t + K) + B_1^T b_{RMHC}^{(t + K)}
\]

\[
= \sum_{u=0}^{T} \left\{ -\tilde{W} \tilde{C}^{RMHC} (\tau + Ku - 1)
\right\} + \tilde{C}(t + K) + B_1^T b_{RMHC}^{(t + K)}
\]

\[
= \sum_{u=0}^{T} \left\{ -\tilde{W} \tilde{C}^{RMHC} (\tau + Ku - 1)
\right\} + \tilde{C}(t + K) + B_1^T b_{RMHC}^{(t + K)}
\]

Since, by convention $\tilde{X}(t) = 0$ for all time $t \leq 0$ and $t > T$, we
know $\phi^T (t - K - 1) = -\tilde{W} \tilde{C}^{RMHC} (t - K - 1) = 0$. Because
RMHC simply solves problem $\phi_{RMHC}^{(t)} (t)$ in Eq.(10) instead of
$\phi_{RMHC}^{(t)} (t)$ in Eq.(9) when $t + K > T$, we know $\psi^T (t + K) \left( \frac{T}{K} \right)
$K-1 \equiv [\hat{W}^T - \hat{C}^T (\tau + K \left[ \frac{\tau}{K} \right] + K)]^T \hat{X}_{\text{RMHC}(\tau)}(\tau + K \left[ \frac{\tau}{K} \right] + K - 1) = 0$. Thus, by removing the first and last zero-value terms and re-group the tail-terms two-by-two in-between, we have the sum of the tail-terms in (53)

\[
\left\{ \begin{array}{l}
\sum_{u=0}^{\left\lfloor \frac{\tau}{2} \right\rfloor -1} \left\{ - \hat{W}^T \hat{X}_{\text{RMHC}(\tau)}(\tau + Ku - 1)
\right.
\left.+ [\hat{W}^T - \hat{C}^T (\tau + Ku)]^T \hat{X}_{\text{RMHC}(\tau)}(\tau + Ku - 1) \right. \\
\end{array} \right.
\]

Finally, since

\[
[\hat{W}^T - \hat{C}^T (\tau + Ku)]^T \hat{X}_{\text{RMHC}(\tau)}(\tau + Ku - 1) \leq \hat{W}^T \hat{X}_{\text{RMHC}(\tau)}(\tau + Ku - 1),
\]

we get the final result that the sum of tail-terms in (53)

\[
\left\{ \begin{array}{l}
\sum_{u=0}^{\left\lfloor \frac{\tau}{2} \right\rfloor -1} \left\{ - \hat{W}^T \hat{X}_{\text{RMHC}(\tau)}(\tau + Ku - 1) + \hat{W}^T \hat{X}_{\text{RMHC}(\tau)}(\tau + Ku - 1) \right. \\
\end{array} \right.
\]

F PROOF OF LEMMA F.1

**Lemma F.1.** (A sufficient condition for the feasibility of dual variables) If the dual variables $(\hat{\beta}(1 : T), \hat{\theta}(1 : T))$ satisfy

(i)  $\hat{\beta}(t) \geq 0, \hat{\theta}(t) \geq 0$,

(ii) there exists $0 \leq \hat{X}(t) < +\infty$, the vector inequality below holds,

\[
\nabla h_t - B_1^T \hat{\beta}(t) + \hat{\theta}(t) - \hat{\theta}(t + 1) \geq 0,
\]

(iii) $\hat{W}^T - \hat{\theta}(t) \geq 0$,

then, they are feasible to the offline dual optimization problem (14).

These three conditions are similar to constraints (14b-14d) and are obtaining from taking the partial derivative of the Lagrangian of problem (33). Condition (55b) guarantees that the dual must stop decreasing at some finite value $\hat{X}(t)$. Condition (55c) guarantees that the dual always increases in $\hat{Y}(t)$. Thus, due to the convexity, it is impossible for it to decrease to $-\infty$.

**Proof.** Note that the Lagrangian of the offline dual optimization problem in Eq.(14) (with convex service cost $h_t(\tilde{X}(t))$ is

\[
L(\hat{X}(1 : T), \hat{Y}(1 : T), \hat{\beta}(1 : T), \hat{\theta}(1 : T)) = \sum_{t=1}^{T} h_t(\hat{X}(t)) + \sum_{t=1}^{T} \hat{W}^T \hat{Y}(t) + \sum_{t=1}^{T} \hat{\beta}(t) \left[ B_2 \hat{a}(t) - B_1 \hat{X}(t) \right] + \sum_{t=1}^{T} \hat{\theta}(t) \left[ \hat{X}(t) - \hat{X}(t - 1) - \hat{Y}(t) \right].
\]

Collecting all the terms with respect to $\hat{X}(t)$ and $\hat{Y}(t)$, respectively, we can know that the feasible dual variables $(\hat{\beta}(1 : T), \hat{\theta}(1 : T))$ (so that the dual objective value will not be $-\infty$) must satisfy the following three conditions simultaneously for any $t \in [1, T]$,

\[
\hat{\beta}(t) \geq 0, \hat{\theta}(t) \geq 0,
\]

\[
\text{min}_{\tilde{X}(t) \geq 0} \left\{ h_t(\tilde{X}(t)) + \left[ -\hat{\beta}(t) B_1 + \hat{\theta}(t) - \hat{\theta}(t + 1) \right] \cdot \tilde{X}(t) \right\} > -\infty,
\]

\[
\text{min}_{\hat{Y}(t) \geq 0} \left\{ \hat{W}^T - \hat{\theta}(t) \right\} \cdot \hat{Y}(t) \geq -\infty.
\]

Up to now, we can see condition (55a) in Lemma F.1 is the same as constraint (57a). Condition (55c) in Lemma F.1 is a sufficient and necessary condition of constraint (57c). Thus, the key for proving Lemma F.1 is to show that the condition (55b) in Lemma F.1 provides a sufficient condition for the constraint (57b).

The function within the minimum operator in Eq.(57b) is the sum of a convex function $h_t(\tilde{X}(t))$ and a linear function $[ - \hat{\beta}(t) B_1 + \hat{\theta}(t) - \hat{\theta}(t + 1) ] \cdot \tilde{X}(t)$ and thus is convex. We first denote the convex function under the minimum operator in Eq.(57b) by $H_t(\tilde{X}(t))$. Let $\tilde{X}(t)$ be the smallest $\tilde{X}(t)$ that condition (55b) in Lemma F.1 holds. We first show that $\tilde{X}(t)$ is the minimizer of the convex function $H_t(\tilde{X}(t))$. Due to the convexity of $H_t(\tilde{X}(t))$,

(i) for all $\tilde{X}(t) > \tilde{X}(t)$, $\nabla H_t(\tilde{X}(t)) \geq 0$, so $H_t(\tilde{X}(t)) \geq H_t(\tilde{X}(t))$;

(ii) if there exists $0 \leq \tilde{X}(t) < \tilde{X}(t)$, then $\nabla H_t(\tilde{X}(t)) < 0$, so $H_t(\tilde{X}(t)) < H_t(\tilde{X}(t))$.

Thus, we know $\tilde{X}(t) = \text{arg min}_{\tilde{X}(t) \geq 0} H_t(\tilde{X}(t))$. Then, we show that the value of $H_t(\tilde{X}(t))$ at $\tilde{X}(t)$ will not be $-\infty$ by contradiction. If $H_t(\tilde{X}(t)) = -\infty$,

(i) for all $\tilde{X}(t) > \tilde{X}(t)$, due to the continuity and convexity of $H_t(\tilde{X}(t))$, for any $\lambda$ such that $0 < \lambda < 1$,

\[
H_t(\lambda \tilde{X}(t) + (1 - \lambda) \tilde{X}(t)) \leq \lambda h_t(\tilde{X}(t)) + (1 - \lambda) h_t(\tilde{X}(t)) = -\infty.
\]

So $H_t(\tilde{X}(t)) = -\infty$, for all $\tilde{X}(t) > \tilde{X}(t)$.

(ii) if there exists $\tilde{X}(t) < \tilde{X}(t)$, then similarly to Eq.(58), we can get for all $\tilde{X}(t) < \tilde{X}(t)$, $H_t(\tilde{X}(t)) = -\infty$.

Thus, $H_t(\tilde{X}(t)) \equiv -\infty$ for all $\tilde{X}(t) \geq 0$. This contradicts the fact that $H_t(0) = h_t(0) \geq -\infty$. Hence, min$_{\tilde{X}(t) \geq 0} h_t(\tilde{X}(t)) + [ -\hat{\beta}(t) B_1 + \hat{\theta}(t) - \hat{\theta}(t + 1) ] \cdot \tilde{X}(t) > -\infty$.

\[
\text{G PROOF OF LEMMA 3.2 WITH CONVEX SERVICE COSTS}
\]

For each round of RMHC(\tau) from time $t$ to $t + K - 1$, where $t = \tau + Ku$ and $u = -1, 0, ..., \left[ \frac{\tau}{K} \right]$, the new formulation of the primal cost

\[
\text{Cost}_{\text{RMHC}(\tau)}(t : t + K - 1) = \sum_{s=t}^{t+K-1} \left\{ h_t(\hat{X}_{\text{RMHC}(\tau)}(s)) + \hat{W}^T \hat{X}_{\text{RMHC}(\tau)}(s) \right\}
\]

(59)
Similarly, the online dual cost $D^{RMHC^{(r)}}(t : t + K - 1)$
\[
= \sum_{s=t}^{t+K-1} \left( \tilde{p}^{RMHC^{(r)}}(s) \right) T B_2 \tilde{A}(s) + \sum_{s=t}^{t+K-1} h_t(\tilde{X}^{RMHC^{(r)}}(t)) \\
+ \sum_{s=t}^{t+K-2} \left[ -B_1^T \tilde{p}^{RMHC^{(r)}}(t) + \tilde{p}^{RMHC^{(r)}}(t) - \tilde{p}^{RMHC^{(r)}}(t + 1) \right]^T \\
\cdot \tilde{X}^{RMHC^{(r)}}(t) + \sum_{s=t}^{t+K-2} \left[ -B_1^T \tilde{p}^{RMHC^{(r)}}(t + K - 1) + \tilde{p}^{RMHC^{(r)}}(t + K - 1) \right]^T \\
\cdot \tilde{X}^{RMHC^{(r)}}(t + K - 1),
\]
(60)
and the tail-terms changes to be
\[
\phi^{(r)}(t - 1) = -\tilde{W}_T \tilde{X}^{RMHC^{(r)}}(t - 1), \\
\psi^{(r)}(t + K - 1) = \left[ \tilde{W}_T - \nabla h_t+B_1^T \tilde{X}^{RMHC^{(r)}}(t + K) ight]^T \\
\cdot \tilde{X}^{RMHC^{(r)}}(t + K - 1),
\]
(61a, 61b)

Then, we still have Lemma 3.2 in Sec. 3.2 holds, i.e., for each $\tau \in [0, K - 1]$ and any $t = \tau + Ku$ where $u = -1, 0, ..., \left\lceil \frac{t}{K} \right\rceil$,
\[
\text{Cost}^{RMHC^{(r)}}(t : t + K - 1) \leq D^{RMHC^{(r)}}(t : t + K - 1) \\
+ \phi^{(r)}(t - 1) + \psi^{(r)}(t + K - 1), \quad \text{for all } \tilde{A}(t : t + K - 1).
\]
(62)

Proof. Similarly as in the proof of Lemma 3.2 in Appendix C, we have complementary slackness conditions that for all time $s \in [t, t + K - 1]$, Eqs.(46a) and (46b) hold. For stationarity/optimality conditions, we only have Eq.(47c) holds. Then, according to Eqs.(46a) and (46b), by adding these zero-value terms, add and deduct a same term
\[
\tilde{W}_T - \nabla h_t+B_1^T \tilde{X}(t+K)=B_1^T B_2 \tilde{A}(t+K)
\]
we have $\text{Cost}^{RMHC^{(r)}}(t : t + K - 1)$
\[
= \sum_{s=t}^{t+K-1} \left( \tilde{p}^{RMHC^{(r)}}(s) \right) T B_2 \tilde{A}(s) + \sum_{s=t}^{t+K-1} h_t(\tilde{X}^{RMHC^{(r)}}(t)) \\
+ \sum_{s=t}^{t+K-2} \left[ -B_1^T \tilde{p}^{RMHC^{(r)}}(t) + \tilde{p}^{RMHC^{(r)}}(t) - \tilde{p}^{RMHC^{(r)}}(t + 1) \right]^T \\
\cdot \tilde{X}^{RMHC^{(r)}}(t) + \sum_{s=t}^{t+K-2} \left[ -B_1^T \tilde{p}^{RMHC^{(r)}}(t + K - 1) + \tilde{p}^{RMHC^{(r)}}(t + K - 1) \right]^T \\
\cdot \tilde{X}^{RMHC^{(r)}}(t + K - 1),
\]
and
\[
\tilde{W}_T - \nabla h_t+B_1^T \tilde{X}(t+K)=B_1^T B_2 \tilde{A}(t+K)
\]
according to Eq.(47c) and using the notations in Eqs.(60) and (61a), (61b), we have $\text{Cost}^{RMHC^{(r)}}(t : t + K - 1)$
\[
\leq D^{RMHC^{(r)}}(t : t + K - 1) - \phi^{(r)}(t - 1) + \psi^{(r)}(t + K - 1).
\]
\hfill \Box

\section{Proof of Theorem 3.5}
We can prove Theorem 3.5 by following the steps for the proof of Theorem 3.1 in Sec. 3.2. First of all, we have verified in Appendix G that Lemma 3.2 holds. Second, according to Lemma F.1, the dual variables $(\tilde{p}^{RMHC^{(r)}}(t), \tilde{p}^{RMHC^{(r)}}(t))$ from each version $\tau$ of RMHC$^{(r)}$ are feasible for the offline dual optimization problem from time $t = \tau + Ku$ to $t + Ku + K - 2$, where $u = -1, 0, ..., \left\lceil \frac{t}{K} \right\rceil$. Then, by introducing the dual variables $(\tilde{p}^{(\tau + (K+1)u + K)} = 0, \tilde{p}^{(\tau + (K+1)u + K)}) =
\tilde{W}_T - \nabla h_t+B_1^T \tilde{X}(t+K)=B_1^T B_2 \tilde{A}(t+K)
\]
for each time $\tau + (K+1)u + K$ between $D^{RMHC^{(r)}}(t : t + (K + 1)u + K - 1)$ and $D^{RMHC^{(r)}}(t : t + (K + 1)u + K - 1)$, where $\tau' = (\tau + (K + 1)u) \text{ mod } K$ and $\tau'' = (\tau + (K + 1)(u + 1)) \text{ mod } K$ in the new group $G^{RMHC^{(r)}}$, we can easily show that Lemma 3.3 in Sec. 3.2 holds following the proof in Appendix D. At last, since
\[
\tilde{W}_T - \nabla h_t+B_1^T \tilde{X}(t+Ku)=B_1^T B_2 \tilde{A}(t+Ku)
\]
by replacing the term $[\tilde{W}_T - \tilde{C}(t+Ku)]^T \tilde{X}^{RMHC^{(r)}}(t + K - 1)$ with
\[
\tilde{W}_T - \nabla h_t+B_1^T \tilde{X}(t+Ku)=B_1^T B_2 \tilde{A}(t+Ku)
\]
in Appendix E, we can see the proof in Appendix E still holds. Now, we can provide the complete proof of Theorem 3.5.
Proof. First, now the service cost is convex and the total cost of ARMHC

\[
\text{Cost}^{\text{ARMHC}}(1 : T) = \sum_{t=1}^{T} h_t(\tilde{X}^{\text{ARMHC}}(t))
\]

\[
+ \sum_{t=1}^{T} \tilde{W}^T [X^{\text{ARMHC}}(t) - \tilde{X}^{\text{ARMHC}}(t-1)]^T.
\]

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

\[
\text{Cost}^{\text{ARMHC}}(1 : T) = \sum_{t=1}^{T} h_t \left( \frac{1}{K} \sum_{i=0}^{K-1} X^{\text{RMHC}(i)}(t) \right)
\]

\[
+ \sum_{t=1}^{T} \tilde{W}^T \left[ \frac{1}{K} \sum_{i=0}^{K-1} X^{\text{RMHC}(i)}(t) - \frac{1}{K} \sum_{i=0}^{K-1} X^{\text{RMHC}(i)}(t-1) \right]^T.
\]

Due to the convexity of \( h_t(\cdot) \) and \( [X^{\text{ARMHC}}(t) - \tilde{X}^{\text{ARMHC}}(t-1)]^T \) and Jensen’s Inequality, we have

\[
\text{Cost}^{\text{ARMHC}}(1 : T) \leq \frac{1}{K} \sum_{t=1}^{T} \sum_{i=0}^{K-1} h_t(X^{\text{RMHC}(i)}(t))
\]

\[
+ \frac{1}{K} \sum_{t=1}^{T} \sum_{i=0}^{K-1} \tilde{W}^T [X^{\text{RMHC}(i)}(t) - X^{\text{RMHC}(i)}(t-1)]^T.
\]

By changing the order of the summations over \( t \) and \( \tau \), and put all rounds of \( \text{RMHC}(i) \) before re-stitching together, we get

\[
\text{Cost}^{\text{ARMHC}}(1 : T) \leq \frac{1}{K} \sum_{t=1}^{T} \left( \sum_{i=0}^{K-1} \sum_{\tau=0}^{t-1} h_t(X^{\text{RMHC}(i)}(t))
\]

\[
+ \tilde{W}^T [X^{\text{RMHC}(i)}(t) - X^{\text{RMHC}(i)}(t-1)]^T \right).
\]

Since Lemma 3.2-3.4 still hold, Eqs.(29-32) still hold. Hence, the competitive ratio of ARMHC in the online optimization problem \((2)\) with convex service costs is still \((\frac{K}{K-1})\)-competitive. \( \square \)

I PROOF OF THEOREM 4.1

Proof. Following the steps from Eq.(25) to Eq.(28) in the proof of Theorem 3.1 in Sec. 3.2, we have

\[
\text{Cost}^{\text{ARMHC-IL}}(1 : T) = \sum_{t=1}^{T} C^T \left( \tilde{X}^{\text{ARMHC-IL}}(t) \right)
\]

\[
+ \sum_{t=1}^{T} \tilde{W}^T [X^{\text{ARMHC-IL}}(t) - \tilde{X}^{\text{ARMHC-IL}}(t-1)]^T
\]

\[
\leq \frac{1}{K} \sum_{t=1}^{T} \sum_{\tau=0}^{t-1} h_t \left( \frac{1}{K} \sum_{i=0}^{K-1} X^{\text{RMHC-IL}(i)}(t) \right)
\]

\[
+ \tilde{W}^T \left[ X^{\text{RMHC-IL}(i)}(t) - X^{\text{RMHC-IL}(i)}(t-1) \right]^T.
\]

Notice that Lemma 3.2 still holds by defining

\[
\text{D}^{\text{RMHC-IL}(i)}(t : t + K - 1) \triangleq \sum_{s=t}^{t+K-1} \left( \hat{\beta}^{\text{RMHC-IL}(i)}(s) \right)^T B_2 A(\cdot),
\]

\[
(65a)
\]

\[
\phi^{(\tau)}(t-1) \triangleq -\tilde{W}^T X^{\text{RMHC-IL}(\tau)}(t-1),
\]

\[
\psi^{(\tau)}(t + K - 1) \triangleq [\tilde{W}^T - C^T (t + K)]^T X^{\text{RMHC-IL}(\tau)}(t + K - 1).
\]

Then, we have \( \text{Cost}^{\text{ARMHC-IL}}(1 : T) \)

\[
\leq \frac{1}{K} \sum_{\tau=0}^{K-1} \sum_{u=1}^{t+K-1} \left( \text{D}^{\text{RMHC-IL}(\tau)}(t + K u - 1) + \phi^{(\tau)}(t + K u - 1) + \psi^{(\tau)}(t + K u + K - 1) \right).
\]

(66)

Similar to Eq.(65a), we define

\[
\text{D}^{\text{RMHC-IL}(i)}(t : t + K - 1) \triangleq \sum_{s=t}^{t+K-1} \left( \hat{\beta}^{\text{RMHC-IL}(i)}(s) \right)^T B_2 A(\cdot).
\]

(67)

From Eq.(65a) and (67), we know that

\[
\text{D}^{\text{RMHC-IL}(i)}(t : t + K - 1) \leq \text{D}^{\text{RMHC-IL}(i)}(t : t + K - 1) = \max_{\{i=0,1,\cdots,K-1,m\in[M]\}} \frac{\bar{a}_m(s)}{a_m(s)}.
\]

(68)

Notice that using the upper of the input only changes the value of the online dual cost from to be \( \text{D}^{\text{RMHC-IL}(i)}(t : t + K - 1) \), but does not affect the constraints of the dual optimization problem. Thus, for the terms involving the online dual cost \( \text{D}^{\text{RMHC-IL}(i)}(t : t + K - 1) \) in Eq.(66), we can apply Eq.(68), the re-stitching idea and Lemma 3.3 described in Sec. 3.2,
tail-terms can still be cancelled. Hence, the total cost of ARMHC-II.

\[
\text{Cost}_{\text{ARMHC-II}}(1 : T) \leq \frac{K + 1}{K} D_{\text{OPT}}(1 : T) \max_{i \in [0, K-1], m \in [1, M]} \frac{\sum_{m} f_{m}(i)}{f_{m}(i)} \leq \frac{K + 1}{K} \max_{i \in [0, K-1], m \in [1, M]} \frac{\sum_{m} f_{m}(i)}{f_{m}(i)} \cdot \text{Cost}_{\text{OPT}}(1 : T)
\]

\[\square\]

**J \quad \text{PROOF OF THEOREM 4.2}**

**Proof.** First, from the cost definition in Eq.(2), we know the total cost of WRMHC

\[
\text{Cost}_{\text{WRMHC}}(1 : T) = \sum_{t=1}^{T} C_{t}(X)_{\text{WRC}} + \sum_{t=1}^{T} W_{t}[X_{\text{WRC}}(t) - X_{\text{WRC}}(t-1)]^{+}
\]

Then, from Eq.(37) that WRMHC takes the weighted average of \( K \) versions of GRMHC decisions and we keep the notation of \( \bar{t}(t, i) \) (t - i) mod \( k \) for any \( i \in [0, k-1] \), we have

\[
\text{Cost}_{\text{WRMHC}}(1 : T) = \sum_{t=1}^{T} C_{t}(X)_{\text{GRMHC}} + \sum_{t=1}^{T} W_{t}[X_{\text{GRMHC}}(t) - X_{\text{GRMHC}}(t-1)]^{+}
\]

Since \( \bar{t}(t, i) = \bar{t}(t, i + 1) \), we have

\[
\sum_{i=0}^{k-1} \rho(i) X_{\text{GRMHC}}(t-1) = \sum_{i=0}^{k-1} \rho(i) X_{\text{GRMHC}}(t-1)
\]

\[
= \sum_{i=0}^{k-1} \rho(i) X_{\text{GRMHC}}(t) + \rho(k-1) X_{\text{GRMHC}}(t-1)
\]

\[
= \sum_{i=0}^{k-1} \rho(i) X_{\text{GRMHC}}(t) + (k-1) X_{\text{GRMHC}}(t-1)
\]

(71a)

Taking the last term \( i = k - 1 \) out from the summation, we get

(71b)

By changing the order of the summations over \( t, i \), and put all rounds of GRMHC\(^{(t)}\) before re-stitching together (See Sec.3.2 for details of re-stitching), we get that

\[
\text{Cost}_{\text{WRMHC}}(1 : T) \leq \frac{1}{k-1} \sum_{i=0}^{k-1} \sum_{t=1}^{T} \sum_{r=0}^{t} \sum_{u=1}^{t} \left( \bar{t}(t-u, i) \right) \rho(t-u) X_{\text{GRMHC}}(r)
\]

\[
+ \sum_{t=1}^{T} W_{t} \left[ X_{\text{GRMHC}}(t) - X_{\text{GRMHC}}(t-1) \right]^{+}
\]

(73)

Notice that Lemma 3.2 still holds by defining

\[
\text{Cost}_{\text{GRMHC}}^{(t)}(t : t + k - 1) \leq \sum_{t=1}^{T} C_{t}(X)_{\text{GRMHC}}^{(t)}(s) + \sum_{t=1}^{T} W_{t} \left[ X_{\text{GRMHC}}^{(t)}(s) - X_{\text{GRMHC}}^{(t)}(s-1) \right]^{+}
\]

(74a)

\[
D_{\text{GRMHC}}^{(r)}(t : t + k - 1) \leq \sum_{s=1}^{r+k-1} \rho(s) X_{\text{GRMHC}}^{(r)}(s)
\]

(74b)

Thus, applying Lemma 3.2 using notations in Eqs.(74a-74d) to Eq.(73), we have that

\[
\text{Cost}_{\text{WRMHC}}(1 : T) \leq \frac{1}{k-1} \sum_{i=0}^{k-1} \sum_{t=1}^{T} \sum_{r=0}^{t} \left( \bar{t}(t-u, i) \right) \rho(t-u) X_{\text{GRMHC}}(r)
\]

(75a)

(75b)

(75c)

(75d)

Eq.(75d) is true due to Lemma 3.4 in Sec.3.2 trivially still holds. From \( Q_{\text{GRMHC}}^{(r)}(t) \) in Eq.(36), we notice that the dual variables from each version \( r \) of GRMHC\(^{(r)}\) satisfy

\[
\rho(s-t) X_{\text{GRMHC}}^{(r)}(s) + \rho(s-t) \bar{C}_{t}^{(r)}(s) - \rho(s-t) \bar{C}_{t}^{(r)}(s)
\]

(76a)

(76b)

(76c)
These dual variables trivially satisfy offline dual constraints (14b-14d) and Lemma 3.3 will still hold. Then, the online dual cost

\[
D_{GRMHC}^{r}(t : t + k - 1 | \tilde{A})
\]

\[
= \sum_{i=t}^{t+k-1} \beta_{GRMHC}^{r}(s) \rho(s-t) B_{2} \tilde{A}(s)
\]

\[
= \sum_{i=t}^{t+k-1} \beta_{GRMHC}^{r}(s) B_{2} \tilde{A}(s) \rho(s-t) \frac{\tilde{A}(s)}{\bar{A}(s)}
\]

\[
\leq \sum_{i=t}^{t+k-1} \beta_{GRMHC}^{r}(s) B_{2} \tilde{A}(s) \rho(s-t) \max_{m \in [1,M]} \tilde{a}_{m}(s)
\]

\[
\leq \sum_{i=t}^{t+k-1} \beta_{GRMHC}^{r}(s) B_{2} \tilde{A}(s) \rho(s-t) \max_{m \in [1,M]} \frac{\sup m(i) - \inf m(i)}{\inf m(i)}
\]

For ease of explanation, the multiplications and divisions of the input vector \( \tilde{A}(t) \) in Eq.(78a) are entry-wise. As mentioned in Sec. 2.2, in Eq.(78b), \( f^{\inf}(0) = 1 \) for all \( n \in [1,N] \) since the current input is already known and there is no uncertainty. Then, applying Eq.(78a), the re-stitching idea and Lemma 3.3 in Sec. 3.2 to the terms involving online dual cost in Eq.(75d), we have

\[
\sum_{i=t}^{t+k-1} \left\{ D_{GRMHC}^{r} \left( \tau + ku : \tau + ku + k - 1 | \bar{A} \right) \right\}
\]

\[
\leq \sum_{i=t}^{t+k-1} \left\{ \sum_{j=1}^{k+1} \max_{i \in [0,k-1], m \in [1,M]} G^{new}_{i,j} \rho(i) \frac{\sup m(i)}{\inf m(i)} \right\}
\]

\[
\leq \sum_{i=t}^{t+k-1} \left\{ \sum_{j=1}^{k+1} \max_{i \in [0,k-1], m \in [1,M]} G^{new}_{i,j} \rho(i) \right\}
\]

\[
\leq (k+1) D_{OPT}^{r}(1 : T) \cdot \max_{i \in [0,k-1], m \in [1,M]} \frac{\sup m(i)}{\inf m(i)} \rho(i).
\]

Hence, the total cost of WRMHC

\[
Cost_{WRMHC}^{r}(1 : T)
\]

\[
\leq \frac{k + 1}{k - 1} D_{OPT}^{r}(1 : T) \cdot \max_{i \in [0,k-1], m \in [1,M]} \frac{\sup m(i)}{\inf m(i)} \rho(i).
\]

Thus, the objective function under the minimizing operator

\[
\min_{\{i \in [0,k-1], m \in [1,M] \}} \max_{i \in [0,k-1]} \frac{f^{up}(i)}{f^{low}(i)} \rho(i)
\]

\[
\leq \frac{k}{k - 1} \sum_{i=0}^{k-1} \rho(i) \cdot Cost_{OPT}^{r}(1 : T).
\]
and the switching-cost coefficient $\tilde{W}$ are generated randomly in advanced. While $\tilde{W}$ is fixed, we allow $\tilde{C}(t)$ to change in time. $\tilde{C}(t)$ are uniformly generated in the continuous interval from 0 to 1, i.e., $\tilde{C}(t) \sim U(0, 1)$. For $\tilde{W} \geq 3\tilde{C}(t)$, $\tilde{W} \geq 20\tilde{C}(t)$ and $\tilde{W} \geq 50\tilde{C}(t)$, we generate $\tilde{W}$ the same way as in the numerical analysis of perfect look-ahead in Sec 5.1.

In Table 2, we compare the empirical performance of WRMHC* with that of CHC using the inputs presented in Counter-example 1 in Sec. 3.1. The values of parameters are provided in Table 1. From [5], 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 be AFHC. (The detailed conditions are provided in [5].) From Table 2, 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 4, we compare the empirical performance of WRMHC* with CHC using the inputs generated from the HP trace at [7] (re-produced in Fig. 5a). Parameters are provided in Table 3. From Table 4, we can observe that, the empirical competitive ratio of WRMHC* is much lower than that of CHC for all cases of different values of switching-cost coefficient. Moreover, when the switching-cost coefficient increases, CHC will get worse since it will pay more costs for switching, especially due to the "end-of-look-ahead" problem we discussed in Sec. 3.1. Contrary to CHC, ECR of WRMHC* decreases as $\tilde{W}$ increases since it will switch less when $\tilde{W}$ becomes larger.