Online Microgrid Energy Generation Scheduling with Accurate Prediction

MENATI, Ali

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Information Engineering

The Chinese University of Hong Kong
January 2020

Abstract of thesis entitled:
Online Microgrid Energy Generation Scheduling with Accurate Prediction
Submitted by MENATI, Ali
for the degree of Master of Philosophy
at The Chinese University of Hong Kong in January 2020

We study the microgrid energy generation scheduling problem with accurate prediction of the near future input. Our goal is to schedule various energy sources in order to minimize the operational cost while satisfying both electricity and heat demand. However, due to uncertainty in the system inputs, including microgrid and renewable generation, designing a scheduling strategy for the problem is substantially different from those of conventional power systems. In the literature, a deterministic online algorithm called CHASE has been proposed to solve this problem. This algorithm achieves a competitive ratio of 3, which is the best possible among deterministic algorithms. Furthermore, this algorithm is extended to utilize the accurate prediction of the near future demand, which improves the competitive ratio of the algorithm. In this thesis, by exploiting the structure of information we propose a new prediction-aware online algorithm that further improves the competitive ratio of the previously best-known algorithm. This algorithm intelligently brings the demand density of the prediction window into consideration and makes more efficient decisions. We use both theoretical analysis and empirical evaluations based on real-world traces to compare our algorithm with the previous ones, and we explain how the

input structure can affect the algorithm performance.

摘要:

本文研究了基于精确短期输入预测的微电网能源生产调度问题。目标是通过调度多种能源来最小化运营成本，同时满足电力和供热需求。但是，基于包括微电网和可再生能源发电量在内的系统输入不确定性，为此类问题设计调度策略的方法与传统电力系统大不相同。现有文献中已有针对此类问题名为CHASE 的确定性在线算法。该算法竞争比为 3，是确定性算法可实现的最优竞争比。此外，该算法通过利用精确短期需求预测信息，已进一步优化竞争比。本文，我们通过发掘预测信息的结构，提出了一种新的基于预测意识的在线算法，使其再次改进已知最优算法的竞争比。新算法智能地评估了预测窗口的需求密度，从而给出更高效的策略。我们通过理论分析和基于真实数据的仿真评估比较了新旧算法，同时阐述了输入信息的结构如何影响算法性能。

Acknowledgement

Throughout the past years of my MPhil study, I have been blessed with lessons and opportunities that have changed my life and provided me with great research experience. My thesis is the result of this journey which ends here, and I want to start my long journey ahead of me by thanking all those who have supported me during these years.

First and foremost, I want to express my sincerest gratitude to my supervisor, Professor Minghua Chen, for his endless support, enthusiastic encouragement, and persistent help throughout my research. He is the one who has made this research possible through his precious guidance and immense knowledge. He has been a source of inspiration and a role model for my academic and research life. From the very beginning, he was the one who believed in me even when I was struggling with the research problems, he was the one who gave me insightful and inspiring lessons and encouragement to continue and helped me overcome all the difficulties step by step. I could not have imagined having a better advisor and mentor for my MPhil study.

I also want to thank Professor Sid Chau for his very helpful suggestions to promote this thesis from various perspectives.

My sincere thanks also go to my fellow lab mates and my dear friends for all their precious support and encouragement during my study. I want to thank Dr. Mohammad Hassan Hajiesmaili, Dr.Hanling Yi, Dr.Qi Chen, Dr. Ying Zhang, Dr. Lei Deng, Qiulin Lin, Jianing Zhai, Tianyu Zhao, Wenjie Xu, Xiang Pan and many others who have been supporting me throughout the time. You are more than friends to me. Thanks for being there for me.

I also want to express my sincere gratitude to my wife for her endless support that made this thesis possible.

Finally, I would like to thank my parents and my family for their warm and endless love and their spiritual support throughout my life. You are the ones who always keep me on the right path.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

A microgrid represents a distributed power system that utilizes both traditional centralized power grid and local generations. Regarded as a promising future power grid paradigm, microgrid can coordinate various energy sources such as renewable energy sources (e.g., wind farms), local generation units (e.g., gas generators), together with the external grid to satisfy the time-varying energy demand of a local community. In recent years, microgrid capacity has witnessed a phenomenal growth of reaching the total global capacity of $3.5GW$ in 2019, and is expected to reach nearly $20GW$ in 2028 at a compound annual growth rate of 21.4% [5]. There are several advantages that make microgrids attractive, including cost efficiency, power reliability, and environmental awareness. Furthermore, an ideal future power grid should be able to integrate more and more renewable energy resources (e.g., wind and solar generation), in order to reduce fossil fuel consumption for the growing environmental concerns. In 2017 renewable energy represented 17.5% of energy consumption in the European Union, which is on the path of the European Commission plan to include 20% renewable energy in the EU energy profile by 2020 [7]. In their latest plan, this target has been raised to reach a share of at least 32% by 2030, which is a significant growth [2].

In microgrid, intelligent energy generation scheduling is a crucial component to achieve a reliable and economical power supply. The scheduling policy should fulfil the time-varying energy demand, by efficiently utilizing different energy sources such as local generators, central grid, and natural gas provider in a way that the total cost of local generation and energy procurement is minimized. This problem is well-studied in the power system literature, and it has appeared in two forms as the economic dispatching [16] and unit commitment [23] problems. In microgrid, however, due to integrating a large fraction of renewable generation, which is highly fluctuating, uncontrollable and intermittent in nature, the scheduling problem becomes more complicated and it requires designing new online scheduling policies [28].

1.2 Thesis Contributions and Organization

In the previous study [26], a deterministic online algorithm called CHASE has been proposed to solve this problem. It is shown that CHASE achieves the competitive ratio of 3, which is the best among all the possible deterministic online algorithms. This algorithm is built upon the structure of the offline algorithm and it tries to track the offline algorithm in an online manner. Furthermore, in [26], a prediction-aware online algorithm is proposed to utilize the accurate prediction of near-term demand or wind forecast. This algorithm can improve the competitive ratio of the prediction-oblivious CHASE. In this thesis, we investigate the prediction-aware online algorithm design for the microgrid cost minimization problem and propose a new prediction-aware online algorithm with an improved competitive ratio over the state-of-the-art. The main contributions of the thesis are summarized as follows:

- We explain why the previous prediction-aware online algorithm can be improved and we give two examples that will clarify the aspects that a new algorithm should take into account in online decision making. We propose CHASEpp as a new prediction-aware online algorithm that can further improve the competitive ratio of CHASElk and makes more efficient online decisions. This online algorithm design space is new in the literature and it is built upon the structure of the offline algorithm and a new parameter that can capture the demand density and its behavior in the prediction window.

- We use both theoretical analysis and trace-driven experiments to evaluate the performance of our algorithm by comparing it with the previous algorithms. We also elaborate on how the input structure and the system parameters can affect the algorithm performance. We note that the idea used in this algorithm can be implemented in the competitive online algorithm design for a general class of problems with a similar nature.

The rest of this thesis is organized as follows. We present related work in Chapter 2. In Chapter 3, problem formulation is presented. Chapter 4 reviews the existing prediction-oblivious online algorithm CHASE and the prediction-aware online algorithm CHASElk. Our proposed prediction-aware online algorithm CHASEpp is described in Chapter 5. The performance analysis and algorithm design strategy is presented in Chapter 6. Extensive experiments are reported in Chapter 7, and finally, Chapter 8 concludes the thesis.

□ **End of chapter.**

Chapter 2

Related Work

In the conventional large-scale power systems, because of the high aggregation effect on demand and small percentage of the erratic renewable generation, it is possible to predict the demand in the whole time horizon with a good level of accuracy and therefore the energy generation scheduling is basically an offline problem, as a result, most of the scheduling policies proposed in these studies are offline solutions. Two main forms of this problem are economic dispatching [15, 31, 16] and unit commitment [23, 17, 30]. The unit commitment problem finds the optimal turn on/off schedule for the large power generation units, and the economic dispatching takes these determined on/off status of the units as its input and schedules the units by changing their output levels. The unit commitment problem is a challenging problem to solve and it is NP-Complete in general [17]. In the literature, the researchers tackled this problem in different ways including dynamic programming [32], stochastic programming [33], and mixed-integer programming [12]. Various heuristics are also proposed to solve the economic dispatching problem [15, 31]. Another direction in economic dispatch is to deploy CHP generators to cover both electricity and heat demand at the same time [18].

In recent years, with the increasing deployment of the highly fluctuating renewable sources and local small scale microgrids, the uncertainty and intermittency have increased substantially on both demand and generation side. Therefore, the previous approaches for the traditional grid are not applicable to this new scenario where we do not know all the information on the time horizon. To overcome this challenge, online solutions are advocated by researchers as a successful alternative. Online optimization has emerged as a foundational topic in a variety of computer systems and it has seen considerable attention from applications in a wide range of researches including networking and distributed systems [9, 29].

In [28], online convex optimization (OCO) framework [37] is used to design algorithms for the microgrid economic dispatch. OCO is a prominent paradigm being increasingly applied in different applications [11, 10, 14, 13]. In [36], a competitive algorithm design approach is used to solve the online economic dispatching problem with peak-based charging model. Both of these studies only consider the economic dispatching problem which does not take into account the start-up cost. The study in [26] incorporates the start-up cost and turns the problem into a joint unit commitment and economic dispatch problem. The paper also takes co-generation into account and proposes a competitive online algorithm with a strong performance guarantee. In [19], a randomized online algorithm is proposed to solve this problem. In this thesis, we aim to solve this problem with accurate prediction of the near future demand. In [26], a prediction-aware online algorithm has been proposed to this end, but it fails to utilize all the given predicted information. Here we propose a new competitive online algorithm that will further improve both theoretical and practical performance over the previous algorithm. Besides the online solution, there have been some studies for the offline energy generation scheduling in microgrid as well [20, 21]. In the recent years, online optimization for data center management has also received a lot of interest from scholars [34, 25, 27, 35].

□ **End of chapter.**

Chapter 3

System Model and Problem Formulation

In the microgrid energy generation scheduling, the objective is to coordinate various energy sources such as local generation units and renewable sources to fulfill both electricity and heat demands while minimizing the total energy cost. It can be formulated as a microgrid cost minimization problem (MCMP), which includes demand covering constraints as well as local generation operational constraints. We consider a system that operates in a time-slotted fashion. Without loss of generality, we assume the length of each time slot is one hour and the total length of the time horizon is T time slots. The key notations are presented in Table 3.1. Next we explain the system model and problem formulation, which are similar to those of [26].

	Notation	Definition
Generator Profile	β	The startup cost of local generator (\$)
	c_{m}	The sunk cost per interval of running local generator (\$)
	c_{o}	The incremental operational cost per interval of running local generator to output an additional unit of power (\$/Watt)
	L	The maximum power output of generator (Watt)
	η	The heat recovery efficiency of co-generation
Demand Profile	$\mathcal{T}$	The set of time slots ($T \triangleq \|\mathcal{T}\|$)
	c_{g}	The price per unit of heat obtained externally using natural gas (\$/Watt)
	$a(t)$	The net electricity demand minus the instantaneous renewable supply at time t (Watt)
	$h(t)$	The heat demand at time t (Watt)
	$p(t)$	The spot price per unit of power obtained from the electricity grid ($P_{\min} \leq p(t) \leq P_{\max}$) (\$/Watt)
	$\sigma(t)$	The joint input at time t: $\sigma(t) \triangleq (a(t), h(t), p(t))$
Opt. Variables	$y(t)$	The on/off status of the local generator (on as "1" and off as "0")
	$u(t)$	The power output level when the generator is on (Watt)
	$s(t)$	The heat level obtained externally by natural gas (Watt)
	$v(t)$	The power level obtained from electricity grid (Watt)

Table 3.1: Key Notations. Brackets indicate the unit. We denote a vector by a single symbol, *e.g.*, $a \triangleq \left[a(t)\right]_{t=1}^{T}$.

3.1 System Model

Energy demand: The energy demand profile includes two types of energy demand, namely electricity demand and heat demand. Let $a(t)$ and $h(t)$ be the net electricity demand (i.e., the residual electricity demand not covered by renewable generation) and the heat demand at time t, respectively. To keep the generality of the problem, we do not assume any specific stochastic model of $a(t)$ and $h(t)$.

External grid and heat: We assume the microgrid operates in the "grid-connected" mode and the unbalanced electricity demand can be acquired from the external grid in an on-demand manner. We denote $p(t)$ as the spot price from the electricity grid at time t, where $p(t) \in [P_{\mathrm{min}}, P_{\mathrm{max}}]$. Again, we do not make any assumption on the stochastic model of $p(t)$. Finally, to cover the heat demand we can use external natural gas and separately generate the heating which costs c_{g} per unit of demand.

Local generation: The power output capacity of the generator is denoted as L. In this thesis, we propose the algorithm for the single generator scenario which could easily be extended to the multiple generator scenario. In [26, Sec. 3.3] it is shown that by correctly partitioning the demand into different layers the case with multiple homogeneous generators can be reduced to the single generator case without suffering any additional performance loss.

By adopting the widely-used generator model [23], We denote β as the startup cost of turning on a generator, c_{m} as the sunk cost per unit time of running a generator in its active state, and c_{o} as the incremental operational cost per unit time for an active generator to output one unit of energy. In a more realistic model of generators, two additional operational constraints are considered. Namely, minimum turning on/off periods, and ramping up/down rates. In [26], a general problem that includes these additional constraints is considered and the approach to solve them is also proposed. In this thesis, we focus on the "fast-responding" generators whose minimum on/off period constraint and ramping-up/down constraint is negligible. Our solution can then be extended to the case with general generators by using the same approach as in [26].

Finally, we assume the local generators are CHP generators that can generate both electricity and heat simultaneously. We denote η as the heat recovery efficiency for co-generation *i.e.*, for each unit of electricity generated, η unit of useful heat can be supplied for free. Thus, ηc_{g} is the cost saving due to using co-generation to supply heat, provided that there is sufficient heat demand. Note that by setting $\eta = 0$ the problem reduces to the case of a system with no co-generation. We assume $c_{\mathrm{o}} \geq \eta \cdot c_{\mathrm{g}}$, which means it is cheaper to obtain heat by using natural gas than purely by generators. To keep the problem interesting, we assume that $c_{\mathrm{o}} + \frac{c_{\mathrm{m}}}{L} \leq p_{\max} + \eta \cdot c_{\mathrm{g}}$. This assumption ensures that the minimum co-generation energy cost is cheaper than the maximum external energy price. If this assumption does not hold, the optimal decision is to always acquire power and heat externally and separately.

3.2 Problem Definition

Let $v(t)$ and $s(t)$ be the amount of electricity and heat obtained by the external grid and the external natural gas, respectively.

Let $y(t)$ be the generator binary on/off status (1 as on and 0 as off), and $u(t)$ be the generator power output level. The microgrid aggregated operational cost over the time horizon $\mathcal{T}$ is given by

$$\mathsf{cost}(y,u,v,s) \triangleq \sum_{t\in\mathcal{T}} \Big\{ \psi\big(\sigma(t), y(t)\big) + \beta[y(t)-y(t-1)]^{+} \Big\},$$

where $\psi\big(\sigma(t), y(t)\big) \triangleq p(t)v(t) + c_{\mathrm{g}}s(t) + c_{\mathrm{o}}u(t) + c_{\mathrm{m}}y(t)$. The operational cost function $\psi\big(\sigma(t), y(t)\big))$ includes both grid and external gas costs (the first two terms), and the operating cost of local generator (the remaining two terms). The total cost is calculated by adding this operational cost along with the generator's switching cost $\beta[y(t)-y(t-1)]^{+}$ over the entire time horizon $\mathcal{T}$. In this thesis, we assume the initial status of the generator is off, *i.e.*, $y(0) = 0$.

Given the cost function and decision variables we formulate the **Microgrid Cost Minimization Problem** (MCMP) as follows:

$$\begin{aligned}
\mathsf{MCMP} \quad & \min_{y,u,v,s} \mathsf{cost}(y,u,v,s) && \text{(3.1a)}\\
\text{s.t.} \quad & u(t) \le L y(t), \quad t\in\mathcal{T}, && \text{(3.1b)}\\
& u(t) + v(t) \ge a(t), \quad t\in\mathcal{T}, && \text{(3.1c)}\\
& \eta u(t) + s(t) \ge h(t), \quad t\in\mathcal{T}, && \text{(3.1d)}\\
\text{vars.} \quad & y(t)\in\{0,1\}, u(t), v(t), s(t) \in \mathbb{R}_0^{+}, t\in\mathcal{T},
\end{aligned}$$

where the packing constraint (3.1b) captures the capacity limit of the generator and the covering constraints (3.1c)-(3.1d) assure the electricity and heat demands are covered using the grid, natural gas, and the generator.

Note that this minimization problem is challenging to solve for several reasons. First, even in offline where the complete knowledge of future information is available this problem is a mixed-integer linear problem, which is generally difficult to solve. Second, the startup cost $\beta[y(t)-y(t-1)]^+$ term in the objective function makes objective function values coupled across the time, hence they cannot be decomposed. Finally, the input profile $\sigma(t) \triangleq (a(t), h(t), p(t))$ which includes the electricity demand, heat demand and the spot price arrives in an online fashion and we do not know the future input, which makes an online solution essential.

□ **End of chapter.**

Chapter 4

Overview of the Existing Solutions

Both online solution and the optimal offline solution for MCMP are presented in [26]. In this chapter, we briefly review these solutions. Their structures are later used in developing our prediction-aware online algorithm.

4.1 Optimal Offline Algorithm Design

As mentioned before, because of the term $\beta[y(t)-y(t-1)]^+$ the objective function values are coupled across the time. But if the on/off status is given, the startup cost is determined and MCMP reduces to a timewise decoupled linear program. The optimal solution of this problem has a closed-form structure which is given in the following lemma

Lemma 1. *[26] Given a fixed on/off status $\big(y(t)\big)_{t=1}^{T}$, the solu-*

tion that minimizes $\textsf{cost}(y, u, v, s)$ *is*

$$u(t)=\begin{cases} 0, & \text{if} \quad p(t)+\eta\cdot c_{\mathrm{g}}\leq c_{\mathrm{o}}, \\ \min\left\{\frac{h(t)}{\eta}, a(t), Ly(t)\right\}, & \text{if} \quad p(t)<c_{\mathrm{o}}<p(t)+\eta\cdot c_{\mathrm{g}}, \\ \min\left\{a(t), Ly(t)\right\}, & \text{if} \quad c_{\mathrm{o}}\leq p(t), \end{cases} \tag{4.1}$$

$$\text{and} \quad v(t)=\left[a(t)-u(t)\right]^{+}, \quad s(t)=\left[h(t)-\eta\cdot u(t)\right]^{+}. \tag{4.2}$$

By the result in Lemma 1, the problem $\textsf{MCMP}$ can be further simplified to following problem $\textsf{sMCMP}$ with a single decision variable to turn on ($y(t)=1$) or off ($y(t)=0$) the generator.

$$\begin{aligned} \textsf{sMCMP} : & \min_{y} \ \textsf{cost}(y) \\ & \text{vars.} \ \ y(t)\in\{0,1\}, t\in\mathcal{T}, \end{aligned}$$

where $\textsf{cost}(y)=\textsf{cost}(y,u,v,s)$ and $(u(t),v(t),s(t))$ are defined based on the result in Lemma 1.

We continue by reviewing the optimal offline solution where the input $\left[\sigma(t)\right]_{t=1}^{T}$ is given at the beginning. Later we use the offline algorithm structure to design our online algorithm. Define

$$\delta(t)\triangleq\psi(\sigma(t),0)-\psi(\sigma(t),1), \tag{4.3}$$

to capture the single-slot cost difference between using or not using the local generation. When $\delta(t)>0$ (resp. $\delta(t)<0$), we tend to turn on (resp. off) the generator. However, because of the startup cost, we should avoid turning on/off the generator too frequently. Hence, to decide when to turn on/off the generator we evaluate whether the *cumulative* gain or loss in the future can compensate for the startup cost or not. Motivated by this issue, the cumulative cost difference $\Delta(t)$ is defined as

$$\Delta(t) \triangleq \min\Big\{0, \max\{-\beta, \Delta(t-1) + \delta(t)\}\Big\}, \tag{4.4}$$

where the initial value is $\Delta(0) = -\beta$. From the definition, one can see that $\Delta(t)$ is computed using only the past and current input and it takes values only within the range $[-\beta, 0]$. Now that $\Delta(t)$ is defined, we start building the critical segments which are later used in the offline algorithm design. According to $\Delta(t)$, we divide the time horizon $\mathcal{T}$ into several disjoint sets called *critical segments*:

$$[1, T_1^c], [T_1^c + 1, T_2^c], [T_2^c + 1, T_3^c], \ldots, [T_k^c + 1, T]$$

where the critical segments are characterized by a set of critical points $T_1^c < T_2^c < ... < T_k^c$, and each type of segment represents similar episodes of demands. Furthermore, each critical slot T_i^c is defined along with an auxiliary slot $\tilde{T}_i^c$, such that the pair $(T_i^c, \tilde{T}_i^c)$ satisfies the following conditions:

- (**Boundary**): Either $\big(\Delta(T_i^c) = 0$ and $\Delta(\tilde{T}_i^c) = -\beta\big)$ or $\big(\Delta(T_i^c) = -\beta$ and $\Delta(\tilde{T}_i^c) = 0\big)$.

- (**Interior**): $-\beta < \Delta(\tau) < 0$ for all $T_i^c < \tau < \tilde{T}_i^c$.

Each pair of $(T_i^c, \tilde{T}_i^c)$ corresponds to an interval where $\Delta(t)$ goes from -β to 0 or from 0 to -β, without touching the boundaries. In Fig. 4.1 one can see an example of dividing the time horizon into critical segments. In [26], it is shown that these critical segments are uniquely defined.

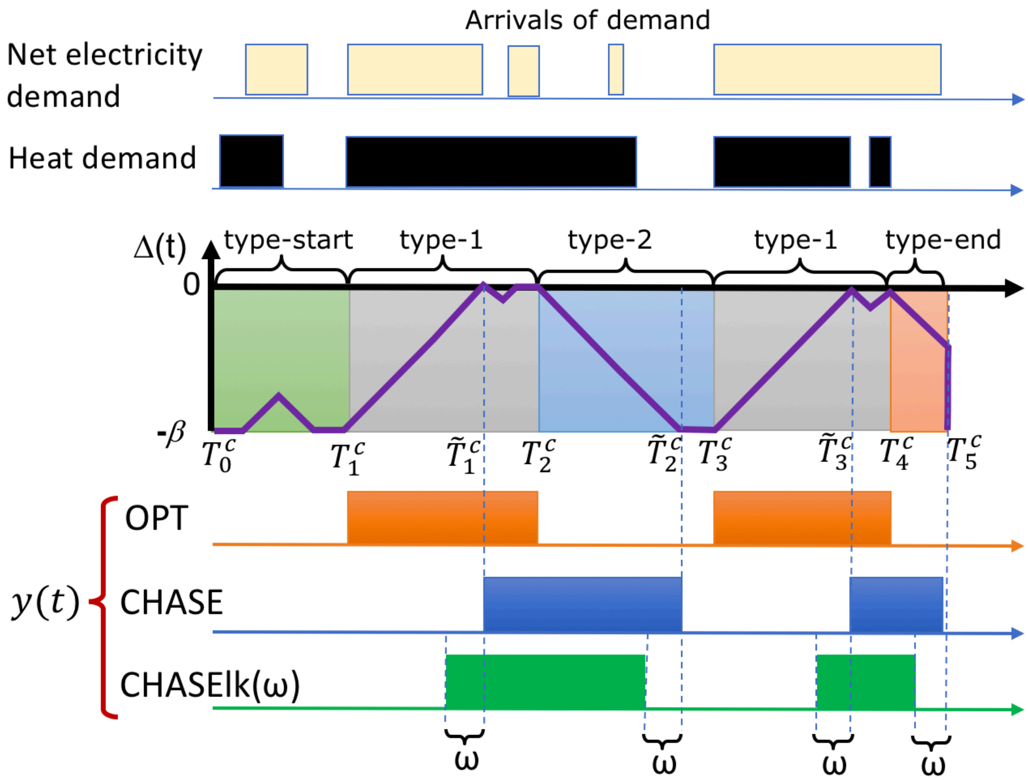

Figure 4.1: An example of $\Delta(t)$ and behaviors of CHASE [26], CHASElk(w), and OPT (that calculates the optimal offline solution).

Now that the time horizon $[1, T]$ is partitioned into critical segments, based on the boundary values of these critical segments we classify them into four categories as follows:

- **type-**0: $[1, T_1^c]$
- **type-**1: $[T_i^c + 1, T_{i+1}^c]$, if $\Delta(T_i^c) = -\beta$ and $\Delta(T_{i+1}^c) = 0$
- **type-**2: $[T_i^c + 1, T_{i+1}^c]$, if $\Delta(T_i^c) = 0$ and $\Delta(T_{i+1}^c) = -\beta$

- **type-**3: $[T_k^c + 1, T]$

By using this definition of the critical segments we proceed to present the optimal solution of sMCMP proposed in [26] in the following theorem.

Theorem 1. *[26] An optimal solution of* sMCMP *is*

$$y^{\star}(t) \triangleq \begin{cases} 1, & \text{if } t \in \textbf{\textit{type-1}}, \\ 0, & \text{otherwise}, \end{cases} \tag{4.5}$$

Finally, to find the optimal offline solution of sMCMP, we obtain the generator on/of status determined by Theorem 1 and use it in Lemma 1 to find the solution. This offline solution is easy to compute and it gives us the key insight to solve this problem in the online setting.

4.2 Online Algorithm Design

To evaluate the performance of the online algorithm the competitive ratio is defined the same as [26]. Let $\mathcal{A}$ be an online algorithm for sMCMP, we have

$$\mathsf{CR}(\mathcal{A}) \triangleq \max_{\sigma} \frac{\mathrm{Cost}(\mathsf{y}_{\mathcal{A}})}{\mathrm{Cost}(\mathsf{y}_{\mathrm{OFA}})} \tag{4.6}$$

Now that competitive ratio is defined we explain the online algorithm CHASE that is proposed in [26].

Recall that in optimal offline solution, right after the process enters type-1 (resp., type-2) critical segment, we set $y(t) = 1$

(resp., $y(t) = 0$). However, in the online setting where no future information is available, the difficulty rises from the fact that it is impossible to immediately detect the beginning of type-1 and type-2, right after entering them.

As shown in Fig. 4.1, although the segment type cannot be immediately determined, fortunately, we can determine it eventually. For example, it is certain that we are in a type-1 critical segment when $\Delta(t)$ eventually reaches 0 for the first time after hitting $-\beta$. By a similar procedure, the other types can also be detected eventually.

Following this observation we present the online algorithm CHASE, which is proposed in [26] and summarized as Algorithm 1. In this algorithm, If $-\beta < \Delta(t) < 0$, CHASE keeps $y(t) = y(t-1)$, since the segment type is not changed. However, when $\Delta(t) = 0$ (resp. $\Delta(t) = -\beta$), we are sure that we entered a type-1 (resp. type-2). Hence, CHASE sets $y(t) = 1$ (resp. $y(t) = 0$). Intuitively, CHASE tracks the offline optimal in an online manner and the online decision changes only after making sure that the offline optimal decision is changed.

Algorithm 1: [26] CHASE, for $t \in \mathcal{T}$

find $\Delta(t)$
if $\Delta(t) = -\beta$ (type-2) **then**
  $y(t) \leftarrow 0$
else if $\Delta(t) = 0$ (type-1) **then**
  $y(t) \leftarrow 1$
else
  $y(t) \leftarrow y(t-1)$
end if
set $u(t)$, $v(t)$, and $s(t)$ according to Eqs. (4.1) and (4.2)

Although CHASE is a simple algorithm, the following theorem shows that it has a strong performance guarantee and it has the best possible competitive ratio among all the possible deterministic online algorithms for the sMCMP problem.

Theorem 2. *[26] The competitive ratio of CHASE satisfies*

$$\mathsf{CR}(\mathit{CHASE}) \leq 3 - 2\alpha, \tag{4.7}$$

where

$$\alpha \triangleq (c_{\mathrm{o}} + c_{\mathrm{m}}/L)/(p_{\max} + \eta c_{\mathrm{g}}) \in (0, 1] \tag{4.8}$$

is a system parameter that represents price discrepancy between using local generation and external sources to supply energy. Furthermore, no other deterministic online algorithm can achieve a better competitive ratio.

4.3 Prediction-Aware Online Algorithm

We consider the setting where at each time slot t the precise prediction of the input for a window of w time slots $\left[\sigma(\tau)\right]_t^{t+w}$, is available. Consider $[T_1^c + 1, T_2^c]$, which is a type-1 critical segment in Fig. 4.1. As we can see when there is no future information available the online algorithm CHASE turns on the generator at time $t = \tilde{T}_1^c$, when $\Delta(t) = 0$ and it knows for sure that the process is in a type-1 critical segment. Now if perfect prediction of the future w time slots is available, by the time $t = \tilde{T}_1^c - w$ the algorithm knows that $\Delta(t)$ will reach zero at time $t = \tilde{T}_1^c$. Hence, the prediction-aware online algorithm can turn on the generator w time slot earlier at time $t = \tilde{T}_1^c - w$.

Algorithm 2: $\mathsf{CHASElk}(w)[t, \sigma(\tau)_{\tau=t}^{t+w}, y(t-1)$

find $\Delta(\tau)_{\tau=t}^{t+w}$
set $\tau' \leftarrow \min\left\{\tau = t, ..., t+w \mid \Delta(\tau) = 0 \text{ or } -\beta\right\}$
if $\Delta(\tau') = -\beta$ (type-2) **then**
  $y(t) \leftarrow 0$
else if $\Delta(\tau') = 0$ (type-1) **then**
  $y(t) \leftarrow 1$
else
  $y(t) \leftarrow y(t-1)$
end if
set $u(t)$, $v(t)$, and $s(t)$ according to Eqs. (4.1) and (4.2)

following this idea, the prediction-aware online algorithm $\mathsf{CHASElk}(w)$ is proposed in [26], and it is summarized in Algorithm 2. We observe that this algorithm is similar to CHASE, except it can detect the critical segment type and turn on/off the generator w time slot ahead of CHASE. Therefore, as shown in Fig. 4.1, even in the worst-case input there is an overlap of size w time slots between online and offline algorithm which means the prediction-aware online algorithm has a better competitive ratio as compared to the prediction-oblivious CHASE.

Theorem 3. *The competitive ratio of the algorithm* $\mathsf{CHASElk}(w)$ *satisfies:*

$$CR = 3 - 2f(\alpha, w) \tag{4.9}$$

where,

$$f(\alpha, w) = \alpha + \frac{(1-\alpha)}{1 + \beta(Lc_{\mathrm{o}} + c_{\mathrm{m}}/(1-\alpha))/(wc_{\mathrm{m}}(Lc_{\mathrm{o}} + c_{\mathrm{m}})}. \tag{4.10}$$

captures the benefit of perfect prediction and monotonically increases from α to 1 as w increases.

This competitive ratio is presented in [26, Sec. 3.2]. In the next chapter, we explain our proposed prediction-aware online algorithm and how it can improve the competitive ratio of the previous algorithm.

□ **End of chapter.**

Chapter 5

New Algorithm Design

In this chapter first, we explain why the previous algorithm needs improvement and then we give examples that shows how a new algorithm can enhance the performance of the previous algorithm.

5.1 Intuitions for Improvement

Consider the examples in Fig. 5.1 and 5.2, for both of these inputs in the type-1 critical segment at time $t = \tilde{T}_1^c - w$ the previous algorithm $\mathsf{CHASElk}(w)$ observes $\Delta(\tilde{T}_1^c) = 0$ and turns on the generator. Recall that $\Delta(t)$ function is the cumulative cost difference between using or not using the generator. When there is a significant amount of demand in the future window, the cost benefit of using the generator over the grid is large. If this value is large enough we tend to turn on the generator, because we know that spending the start up cost is worth it when we can enjoy the cost benefit of using the generator in the coming window. On the other hand, if a small value of demand is coming in every window, we do not need to turn on the generator, because the cost benefit of using the generator is small and using the grid to cover the small value of the incoming demand is better than spending the large start up cost.

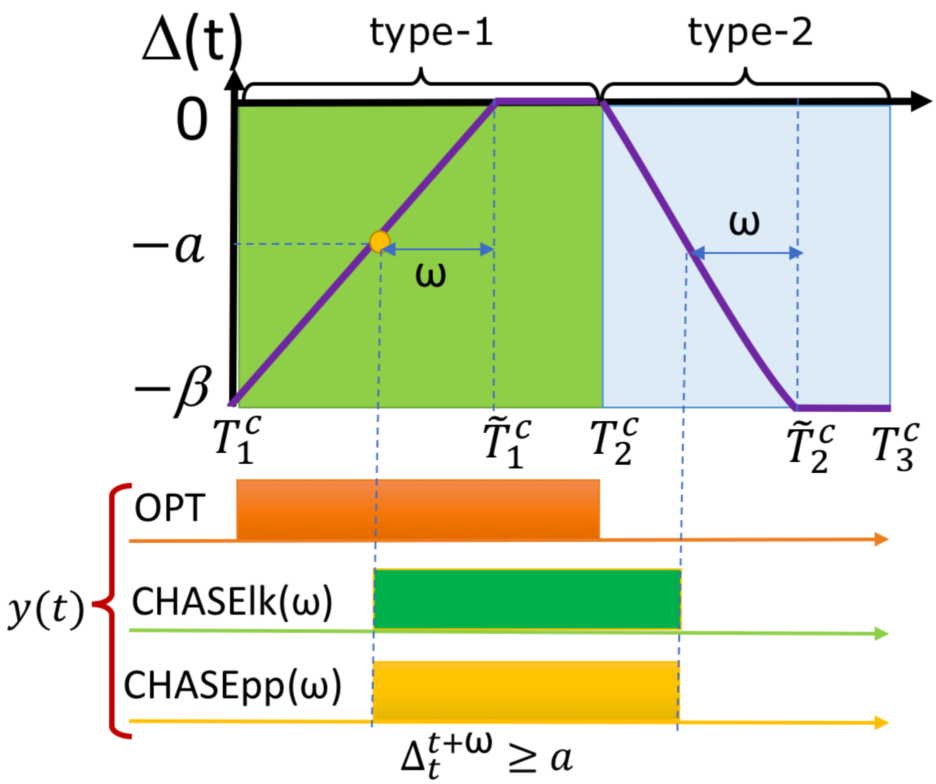

Figure 5.1: Input with $\Delta(\tilde{T}_1^c - w) = -a$, where we should turn on the generator .

We can see from these two figures that for the time interval $[\tilde{T}_1^c - w, \tilde{T}_1^c]$, the cumulative cost difference in Fig. 5.1 is large $(\Delta(\tilde{T}_1^c) - \Delta(\tilde{T}_1^c - w) = a)$, while in Fig. 5.2 the cost difference is almost zero. The idea is that when the perfect prediction of the future window is available, and the segment type is detected ahead of time, to turn on/off the generator we should also take into account the cumulative cost difference in the future window. Using these two examples we explain two intuitions that motivate us to propose a new online algorithm that can capture these differences of the inputs.

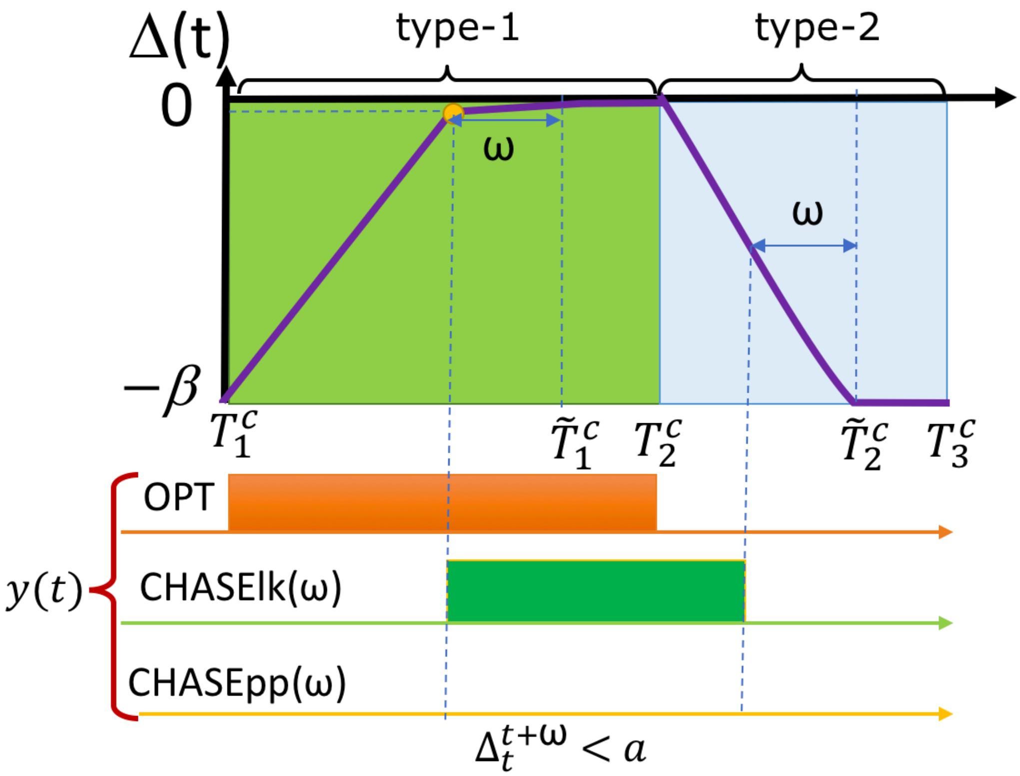

Figure 5.2: Input with $\Delta(\tilde{T}_1^c - w) \approx 0$ where we should keep the generator off.

- The first intuition is that in Fig. 5.2 the cumulative cost difference in the window $[\tilde{T}_1^c - w, \tilde{T}_1^c]$ is almost zero which means there is no difference between using or not using the generator and turning on the generator will not give us any gain in this window. On the other hand, this cumulative cost difference in Fig. 5.1 is large, hence if we turn on the generator at the beginning of this window, we could enjoy a significant gain. Intuitively, if in a window the cost difference is very small it is better to keep the generator off for this window and turn the generator on only if there is a significant cost difference in the future window.

- The second intuition is that in Fig. 5.2 when $\Delta(\tilde{T}_1^c)$ reaches zero $\Delta(\tilde{T}_1^c - w)$ is almost zero and it means we already lost a lot of demands and suffered a lot from not turning on the generator earlier. Hence, turning on the generator and spending the startup cost at the current time $\tilde{T}_1^c - w$ when there is not enough demand in the future window is not beneficial. On the other hand in Fig. 5.2 at time $\tilde{T}_1^c - w$ we have $\Delta(\tilde{T}_1^c - w) = -a$, which means that we have not lost too much demand and turning on the generator at this moment when there is also enough demand in the future window is beneficial.

Following these intuitions, we proceed to explain our online algorithm.

5.2 Cost Difference in the Window

First, to capture the behavior of the input in the future window we define the cumulative cost difference in an interval as follows.

Definition 1. *For any $\tau \in [t, t+w]$, we define Δ_t^τ as the cost difference between using or not using the generator in the time* *interval* $[t, \tau]$*, as follows:*

$$\Delta_t^\tau = \sum_{s=t}^{\tau} \delta(s). \tag{5.1}$$

Recall that $\delta(s)$ is the cost difference between using or not using the generator at the single time slot s. The value of Δ_t^{t+w} represents the density of demand between time t and $t + w$ and helps us in determining when to turn on/off the generator. These intuitions motivate us to consider the cost difference in the window when we want to turn on the generator in a type-1 critical segment.

However, for a type-2 critical segment this is not the case. Consider a type-2 critical segment $[T_2^c + 1, T_3^c]$ in Fig. 5.1, where there is no arriving demand in $[\tilde{T}_2^c, T_3^c]$. The offline algorithm has turned off the generator at the beginning of type-2 critical segment, hence the offline cost in $[\tilde{T}_2^c, T_3^c]$ is zero. At time $\tilde{T}_2^c - w$ the algorithm is able to determine that we are in a type-2 critical segment and at this point if the algorithm keeps the generator on it will cost c_{m} unit per time slot. Hence no matter what is the cumulative cost difference in the window the online algorithm should turn off the generator as soon as it detects the type-2 critical segments. This observation is aligned with our understanding of the type-2 segment which is similar to a ski-rental problem where keeping the generator on, is like the skier keeps renting the ski and its online cost keeps increasing while the offline algorithm has already bought the ski and the offline cost is fixed. Hence, in the type-2 critical segment we do not use Δ_t^{t+w} to make the online decision. In the following section we describe the algorithm in more details.

5.3 Algorithm Description

In this algorithm, we wait until we make sure we are in the type-1 critical segment and then we check the future window and if the cost difference between using or not using the generator is large enough in this window we turn on the generator, otherwise we keep the generator off. The key idea here is that if in a window the cost difference between using or not using the generator is small it means the difference between online and offline algorithm is also small which means we do not need to spend the startup cost β, and by keeping the generator off the competitive ratio also stays small.

Algorithm 3: $\mathsf{CHASEpp}(w)[t, \sigma(\tau)_{\tau=t}^{t+w}, y(t-1)$

1: find $\Delta(\tau)_{\tau=t}^{t+w}$
2: set $\tau' \leftarrow \min\left\{\tau = t, ..., t+w \mid \Delta(\tau) = 0 \text{ or } -\beta\right\}$
3: **if** $\Delta(\tau') = -\beta$ (type-2) **then**
4:     $y(t) \leftarrow 0$
5: **else if** $\Delta(\tau') = 0$ (type-1) **then**
6:     **if** $\Delta_t^{t+w} \geq a$ or $\Delta_t^{\tau'} \geq \beta$ **then**
7:         $y(t) \leftarrow 1$
8:     **end if**
9: **else**
10:     $y(t) \leftarrow y(t-1)$
11: **end if**
12: set $u(t)$, $v(t)$, and $s(t)$ according to Eqs. (4.1) and (4.2)

This prediction-aware online algorithm is summarized in Algorithm 3. In line 3 of the algorithm, one can see that if we know we are in a type-2 critical segment we turn off the generator without considering Δ_t^{t+w}. But if we know we are in a type-1 critical segment (lines 5 to 8) we have two cases.

- In the first case, we have $\Delta_t^{t+w} \geq a$ which means there exist enough demand in the future window to turn on the generator so we will turn on the generator.
- In the second case, we have $\Delta_t^{\tau'} \geq \beta$ which means between time t and τ' there is β unit of cost difference that can compensate the startup cost, hence we can easily turn on the generator at time t.

If none of these two cases happens we do not turn on the generator and we just keep the generator status as before. By comparing the description of this algorithm with Algorithm 2, we observe that their difference is that after detection the type-1 critical segment, Algorithm 2 always turns on the generator, when our new algorithm turns on the generator only if it makes sure there is enough demand in the future window ($\Delta_t^{t+w} \geq a$). This means that our new algorithm takes both the segment type, and the slope of the future window into account when making online decisions.

The online algorithm design space used in this thesis is new in the literature. In this design space turning on the generator depends on satisfying two conditions at the same time. The first condition is that we make sure the offline algorithm has turned on the generator so we are following the offline solution and the second condition is that we turn on the generator only if there is a significant value of demand in the future window that is worth turning on the generator.

We note that the previous algorithm CHASElk(w), is a simple extension of CHASE where we just track the offline solution in an online manner and as soon as we know the offline algorithm has turned on the generator, the online algorithm turns on the generator. But our proposed online algorithm tracks the offline algorithm in a smart manner and it does not exactly imitate the behaviour of the offline algorithm. For example consider a time where the online algorithm knows that offline algorithm has turned on the generator, but at that time the online algorithm has already lost a significant amount of demand and the demand density in the future window is also small. In this situation turning on the generator and spending the large start up cost is not a good decision and it is better to keep the generator off. By using this new and effective strategy the algorithm becomes more competitive and it makes more smart decision rather than simply tracking the offline algorithm. This new and effective design space can improve the performance and the competitive ratio of the previously best known algorithm.

Now the algorithm is described and we need to compute the optimal value of the threshold. In the next chapter, we analyze how the performance of the new algorithm changes as a function of the threshold a and we compute the optimal threshold.

□ **End of chapter.**

Chapter 6

Performance Analysis

In this chapter, first, we calculate the competitive ratio of the algorithm and then we choose the threshold by analyzing this competitive ratio.

6.1 Competitive Ratio Analysis

To find the optimal value for the threshold, we first calculate the competitive ratio as a function of the threshold a. Let us denote the algorithm with a as its threshold to be $\mathcal{A}(a)$. We define the competitive ratio of this algorithm as $\mathsf{CR}(\mathcal{A}(a))$. We have:

Lemma 2. *The competitive ratio of Algorithm 3 with a is*

$$\mathsf{CR}(\mathcal{A}(a)) = \max\{R_{\text{on}}(a), R_{\text{off}}(a)\} \tag{6.1}$$

where $R_{\text{on}}(a)$ is a decreasing function given by

$$R_{\text{on}}(a) = 1 + \Big(1 - \frac{Lc_{\text{o}} + c_{\text{m}}}{L(p_{\max} + \eta \cdot c_{\text{g}})}\Big) \cdot \max_{q \in \{0, wc_{\text{m}}\}} \tag{6.2}$$
$$\Big\{\frac{2\beta - q}{\beta + \big(2wc_{\text{m}} - q + \frac{c_{\text{o}}}{p_{\max}+\eta\cdot c_{\text{g}}}a\big)\big(1 - \frac{c_{\text{m}}}{L(p_{\max}+\eta\cdot c_{\text{g}} - c_{\text{o}})}\big)}\Big\},$$

and $R_{\text{off}}(a)$ is an increasing function given by

$$R_{\text{off}}(a) = \frac{(w+1)c_{\text{m}} + a}{(w+1)c_{\text{m}} + \frac{c_{\text{o}}}{p_{\max}+\eta\cdot c_{\text{g}}}a}. \tag{6.3}$$

Proof. Refer to Appendix 9.1. □

To calculate the competitive ratio of the algorithm, we need to see how the algorithm performs for different inputs and find the worst performance ratio of the algorithm among all the possible inputs. For this algorithm, we observe that depending on the value of a this algorithm has two different types of the worst-case input. We denote the performance ratio of the algorithm for these two possible worst-case inputs $R_{\text{on}}(a)$ and $R_{\text{off}}(a)$. Therefore, the competitive ratio can be found by finding the maximum value among these two possible values. In this section, we explain the competitive ratio of this new algorithm and how to find this competitive ratio. Here we explain both of these worst-case inputs and their corresponding performance ratios.

6.1.1 First Worst-Case Input R_{on}

Consider the input shown in Fig. 6.1, where $[T_i^c + 1, T_{i+1}^c]$ and $[T_{i+1}^c + 1, T_{i+2}^c]$ are a type-1, and type-2 critical segments, re-

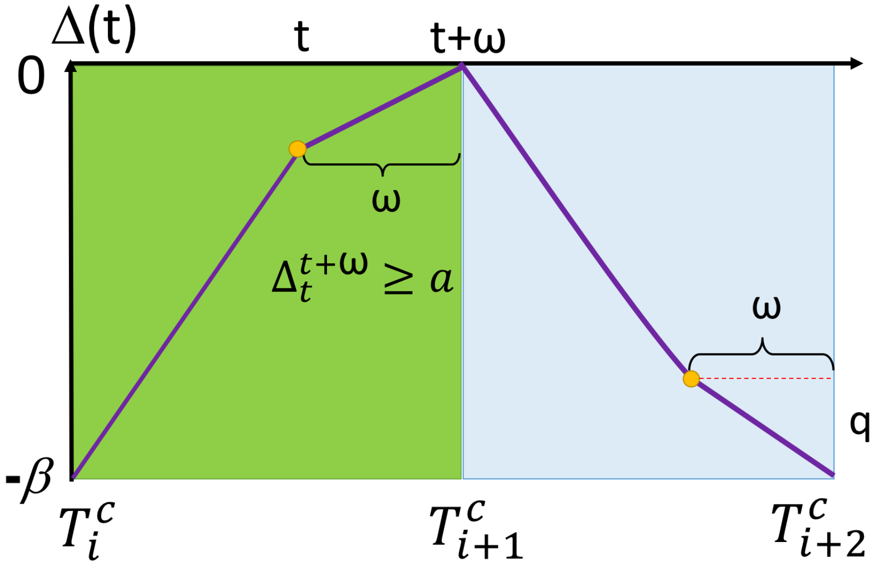

Figure 6.1: The worst case input for the type-1 and type-2 critical segments.

spectively. We assume in the type-1 critical segment at time $t = T^c_{i+1} - w$, the algorithm observes $\Delta_t^{t+w} \geq a$, and turns on the generator, and in the type-2 critical segment at time $t = T^c_{i+2} - w$, the algorithm observes $\Delta(t+w) = -\beta$, and turns off the generator. We calculate performance ratio of the algorithm for this input as follows:

$$\frac{\mathrm{Cost}(y_{\mathsf{CHASEpp}(w)})}{\mathrm{Cost}(y_{\mathrm{OFA}})} \leq 1 + \left(1 - \frac{Lc_{\mathrm{o}}+c_{\mathrm{m}}}{L(p_{\max}+\eta\cdot c_{\mathrm{g}})}\right) \cdot \max_{q \in \{0, wc_{\mathrm{m}}\}} \left\{ \frac{2\beta - q}{\beta + \left(2wc_{\mathrm{m}} - q + \frac{c_{\mathrm{o}}}{p_{\max}+\eta\cdot c_{\mathrm{g}}} a\right)\left(1 - \frac{c_{\mathrm{m}}}{L(p_{\max}+\eta\cdot c_{\mathrm{g}} - c_{\mathrm{o}})}\right)} \right\} \tag{6.4}$$

where $q = \Delta(T^c_{i+2} - w) - \Delta(T^c_{i+2})$ is shown in the picture. This performance ratio is calculated by finding the maximum value across all possible values of q. We denote this performance ratio $R_{\mathrm{on}}(a)$. From this performance ratio, we observe that the worst case in type-2 depends on the value of w and it is one of the two cases $q \in \{0, wc_{\mathrm{m}}\}$. If the prediction window size is small, $q = wc_{\mathrm{m}}$ is the worst case and if the prediction window size is large the worst case is $q = 0$. This makes sense because we know we turn off the generator at $t = T^c_{i+2} - w$ and at this time we have $\Delta(T^c_{i+2} - w) = -\beta + q$. If w is large, for example $wc_{\mathrm{m}} = \beta$, then for $q = wc_{\mathrm{m}}$ we have $\Delta(T^c_{i+2} - w) = -\beta + q = 0$ which means the algorithm turns off the generator in the beginning of the type-2 critical segment and this clearly can not be the worst case. Hence for large w the worst case is $q = 0$.

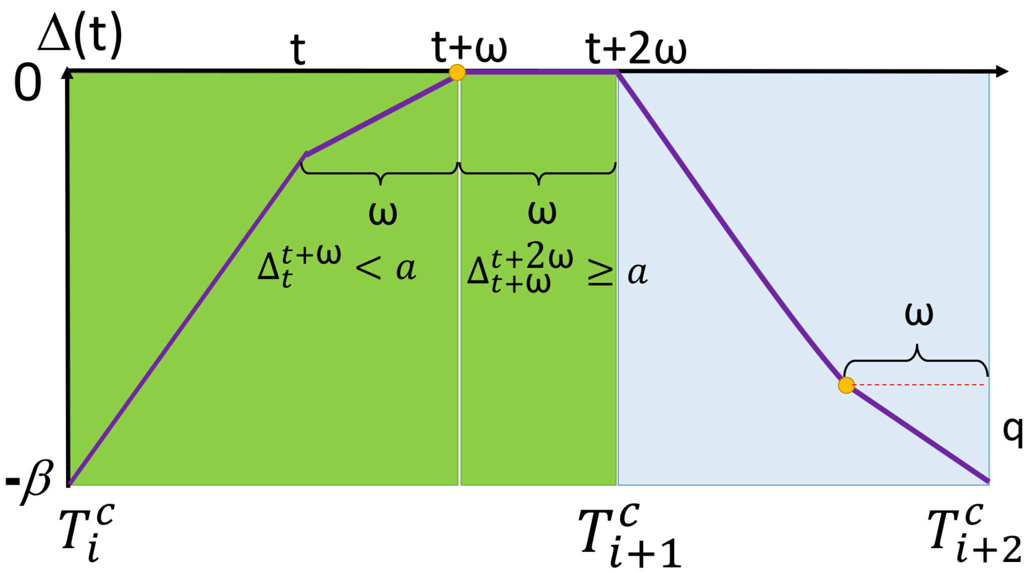

Figure 6.2: Performance change by keeping the generator off for a window.

6.1.2 Second Worst-Case Input R_{off}

Now we want to explain the second type of the possible worst-case input. Consider the example in Fig. 6.2, where at time t we have $\Delta_t^{t+w} < a$ and the algorithm keeps the generator off, but at time $t + w$ we have $\Delta_{t+w}^{t+2w} \geq a$ and the algorithm turns on the generator. We observe that this input is similar to the previous one, where the only difference is that this new input has an additional window with $\Delta_t^{t+w} < a$. We want to see how the offline and online cost changes compared to the previous performance ratio. In this example, both online cost and offline cost will increase. We introduce the following lemmas which give an upper bound on the online cost over the offline cost for such a window.

Lemma 3. *Consider a window* $[t, t+w]$*, in the type-1 critical segment. If we have* $\Delta_t^{t+w} \leq a$*, then the cost of the online algorithm over the cost of the optimal offline algorithm in this window is upper bounded by the following:*

$$\frac{\text{Cost}(y_{\mathsf{CHASEpp}(w)})}{\text{Cost}(y_{\text{OFA}})} \leq \frac{(w+1)c_{\text{m}} + a}{(w+1)c_{\text{m}} + \frac{c_{\text{o}}}{p_{\max}+\eta\cdot c_{\text{g}}} a}. \tag{6.5}$$

Proof. Refer to Appendix 9.2. □

As one can see from this lemma the online and offline cost increment is as follows:

$$\frac{\text{online cost increment}}{\text{offline cost increment}} \leq \frac{(w+1)c_{\text{m}} + a}{(w+1)c_{\text{m}} + \frac{c_{\text{o}}}{p_{\max}+\eta\cdot c_{\text{g}}} a}. \tag{6.6}$$

we denote this value in (6.6) as $R_{\text{off}}(a)$. If this value is larger than than the first performance ratio $R_{\text{off}}(a) \geq R_{\text{on}}(a)$, then by adding this window the ratio increases and by adding infinite number of windows it approaches $R_{\text{off}}(a)$. On the other hand, if we have $R_{\text{off}}(a) < R_{\text{on}}(a)$ the ratio will not increase and the competitive ratio is still less than $R_{\text{on}}(a)$. Hence, in both case the competitive ratio is equal to maximum value between $R_{\text{on}}(a)$ and $R_{\text{off}}(a)$. This explains the intuition behind the competitive ratio.

6.2 The Optimal Threshold

Now that we calculated $\mathsf{CR}(\mathcal{A}(a))$, we want to find the algorithm with the minimum competitive ratio. In other words, we need to find the optimal a^* that minimizes the competitive ratio

$$a^* = \arg\min_a CR(\mathcal{A}(a)) = \arg\min_a \max\{R_{\text{on}}(a), R_{\text{off}}(a)\} \quad (6.7)$$

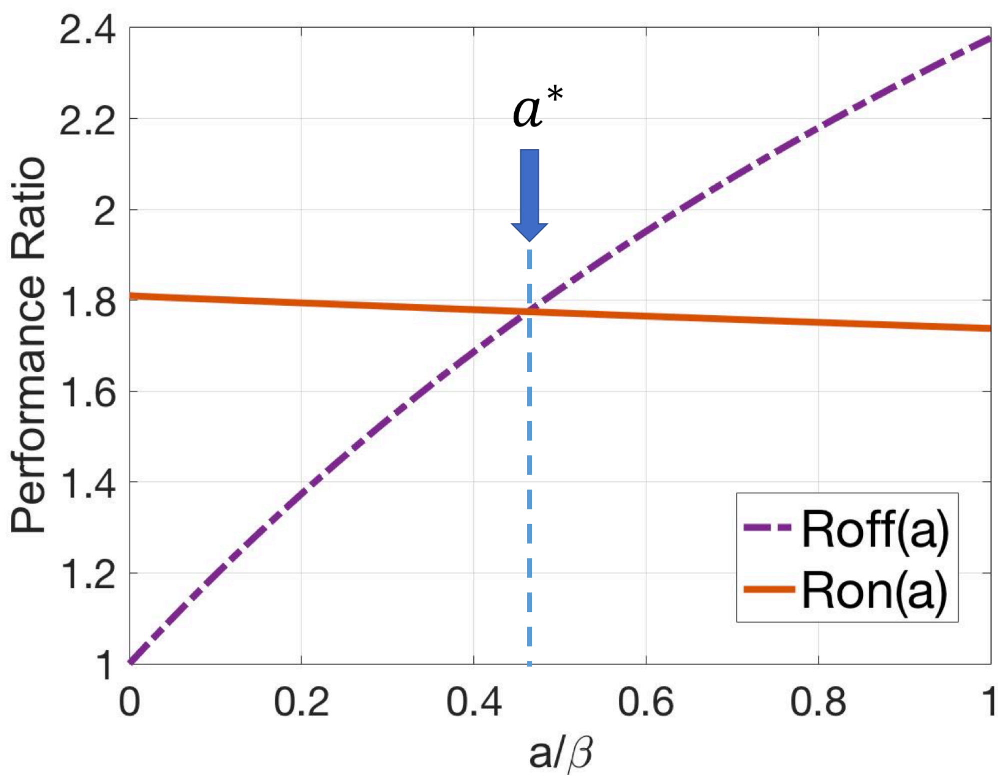

Figure 6.3: The value of a can be found by finding the intersection of $R_{\text{on}}(a)$ and $R_{\text{off}}(a)$.

To understand how to find a^* and its correspondent CR, we explain the behavior of $R_{\rm off}(a)$, and $R_{\rm on}(a)$. Consider the example given in Fig. 6.3. In this example we observe that at $a = 0$, we have $R_{\rm off}(0) \leq R_{\rm on}(0)$, on the other hand, by increasing a, $R_{\rm on}(a)$ always decreases while $R_{\rm off}(a)$ always increases. Hence, the optimal value of a can be found by finding the point that $R_{\rm off}(a)$ reaches $R_{\rm on}(a)$ for the first time.

From Lemma 2 we know that $R_{\rm on}(a)$ is always a decreasing function and $R_{\rm off}(a)$ is always an increasing function. Therefore we can always find the optimal value as follows

$$
\begin{aligned}
& a^* = \max a && (6.8a)\\
\text{s.t.}\quad & 0 \leq a \leq \beta, && (6.8b)\\
& 0 \leq a \leq L\big(p_{\max} + \eta \cdot c_{\rm g} - c_{\rm o} - \frac{c_{\rm m}}{L}\big)w, && (6.8c)\\
& R_{\rm on}(a) \geq R_{\rm off}(a), && (6.8d)
\end{aligned}
$$

where (6.8b) ensures the threshold is not larger than the startup cost. This makes sense because if the cost difference is larger than the startup cost it is always optimal to turn on the generator. The constraint (6.8c) ensures that the threshold is within the maximum possible value for the cost difference in a window with size w, and the last constraint (6.8d) ensure that we find the threshold that gives us the minimum competitive ratio. This optimization problem is easy to solve. After finding the feasible interval for a by using (6.8c) and (6.8d), we just need to find the intersection point of two functions $R_{\rm on}(a)$ and $R_{\rm off}(a)$. Since $R_{\rm off}(a)$ is always increasing and $R_{\rm on}(a)$ is always decreasing, we can find their intersection by using a simple binary search.

Now that the optimal threshold is calculated, we present the competitive ratio of our proposed online algorithm:

Theorem 4. *The competitive ratio of the algorithm* $\mathsf{CHASEpp}(w)$ *satisfies:*

$$CR = 3 - 2g(\alpha, w) \tag{6.9}$$

where,

$$g(\alpha, w) = \alpha + (1-\alpha)\Big(1 - \frac{1}{2} \cdot \max_{q \in \{0, wc_{\mathrm{m}}\}} \Big\{ \frac{(2\beta - q)}{\beta + \big((2wc_{\mathrm{m}} - q)(Lc_{\mathrm{o}} + c_{\mathrm{m}}) + \alpha Lc_{\mathrm{o}} a^*\big)/\big(Lc_{\mathrm{o}} + c_{\mathrm{m}}/(1-\alpha)\big)} \Big\}\Big) \tag{6.10}$$

captures the benefit of perfect prediction and monotonically increases from α *to* 1 *as* w *increases.*

Proof. Refer to Appendix 9.3. □

Note that a^* is obtained by solving (6.8). The competitive ratio in (6.9) depends on the value of the window w and the optimal threshold a^*. In particular, if there is no prediction ($w = 0$), we have $g(\alpha, 0) = \alpha$, which gives us the competitive ratio of CHASE without prediction. On the hand, if w is large the value in (6.10) reduces to the following:

$$g(\alpha, w) = \alpha + \frac{(1-\alpha)}{1 + \beta(Lc_{\mathrm{o}} + c_{\mathrm{m}}/(1-\alpha))/(2wc_{\mathrm{m}}(Lc_{\mathrm{o}} + c_{\mathrm{m}}) + \alpha Lc_{\mathrm{o}} a^*)}. \tag{6.11}$$

We can see that this function depends on the value of the window size and the optimal threshold, and by increasing the value of a^* or w value of the function $g(\alpha, w)$ keeps increasing which means the competitive ratio keeps decreasing. By comparing this function with the competitive ratio of $\mathsf{CHASElk}(w)$ presented in (4.9) we can see that the our new algorithm $\mathsf{CHASEpp}(w)$ always has a better competitive ratio compared to the previous best known algorithm $\mathsf{CHASElk}(w)$.

Note that from the definition of the optimal threshold a^* we have $R_{\text{on}}(a^*) \geq R_{\text{off}}(a^*)$. Therefore for the competitive ratio we have $CR = R_{\text{on}}(a^*)$. On the other hand, the maximum value of $R_{\text{off}}(a)$ is equal to $1/\alpha$, which is equal to the competitive ratio of the algorithm that never turns on the generator. In other words, if we have $CR = R_{\text{on}}(a^*) \geq 1/\alpha$, it means the algorithm that never turns on the generator has a better competitive ratio compared to our algorithm and it is better not to use our algorithm. Instead we should keep the generator off and never turn on the generator. We summarize this result in the following algorithm which compares the competitive ratio of our algorithm with $1/\alpha$ and uses our algorithm if it has a smaller competitive ratio.

Algorithm 4: $\mathsf{CHASEpp}^+(w)[t, \sigma(\tau)_{\tau=t}^{t+w}, y(t-1)]$

```
1: if  1/α < CR(CHASEpp(w))  then
2:     y(t) ← 0,  u(t) ← 0,  v(t) ← a(t),  s(t) ← h(t)
3:     return (y(t), u(t), v(t), s(t))
4: else
5:     return CHASEpp(w)[t, σ(τ)_{τ=t}^{t+w}, y(t − 1)]
6: end if
```

Corollary 1. *The competitive ratio of* $\mathsf{CHASEpp}^+(w)$ *satisfies:*

$$CR(\mathsf{CHASEpp}^+(w)) = \min\{CR(\mathsf{CHASEpp}(w)), \frac{1}{\alpha}\} \tag{6.12}$$

By using the same logic $\mathsf{CHASElk}^+(w)$ is presented in [26]. In the rest of this thesis, we evaluate the performance of this new algorithm.

□ **End of chapter.**

Chapter 7

Empirical Evaluations

We carry out numerical experiments using real-world traces to evaluate the performance of our proposed online algorithm compared to the previous algorithm. To capture the performance of the algorithm, we calculate the cost incurred by using only external electricity, heating and wind energy (when no generator is utilized) as a benchmark and we report the cost reduction of different algorithms as compared to this benchmark. The length of each time slot is 1 hour and the total cost incurred during one week (T = 168) is reported.

7.1 Experiment Setting

Data traces and system model: We obtain the electricity and heat demand traces from [1]. These traces belong to a college in San Francisco, with yearly electricity consumption of around $154GWh$, and gas consumption of around 5.1×10^6 therms. We use wind power traces from [4] to calculate the net electricity demand. These power output data are from a wind farm outside San Francisco with an installed capacity of $12MW$. We obtain the electricity and natural gas price data from PG&E [6], where depending on the daily hours the electricity price takes one of the three possible on-peak, mid-peak, or off-peak tariffs. We deploy generators with the same specifications as the one in [8], with heat recovery efficiency η set to be 1.8, and the maximum capacity of $L = 3MW$. According to the natural gas price and the generator efficiency, the incremental cost c_{o} and running cost c_{m} per unit time are set to be $\$0.051/KWh$ and $\$110/h$ respectively. We consider a heating system with the unit heat generation cost of $c_{\mathrm{g}} = \$0.0179/KWh$, according to [3]. The startup cost β is set to be \$1400, which is equivalent to running the generator at its full capacity for about 5 hours at its own operating cost. The peak for the electricity demand is $30MW$, so we use 10 homogeneous generators to fully satisfy the demand. Recall that in Sec. 3.1, we explained that our algorithm is proposed in the single generator case but it can be extended to the multiple-generator case without performance loss.

Comparison algorithms: We compare the performance of the following algorithms:

- The optimal offline solution OPT, which operates in the presence of the full knowledge of the input over the whole time horizon $\mathcal{T}$.

- The online algorithm CHASE, which is proposed in [26] and is summarized in Algorithm 1.

- The prediction-aware online algorithm $\mathsf{CHASElk}^{+}(w)$, which is proposed in [26] and is explained in Chapter 4.

- The prediction-aware online algorithm $\mathsf{CHASEpp}^+(w)$, which is proposed in this thesis as Algorithm 4 and is explained in Chapter 5.

- The prediction-aware online algorithm RHC, which is a popular algorithm widely-used in the control literature [24].

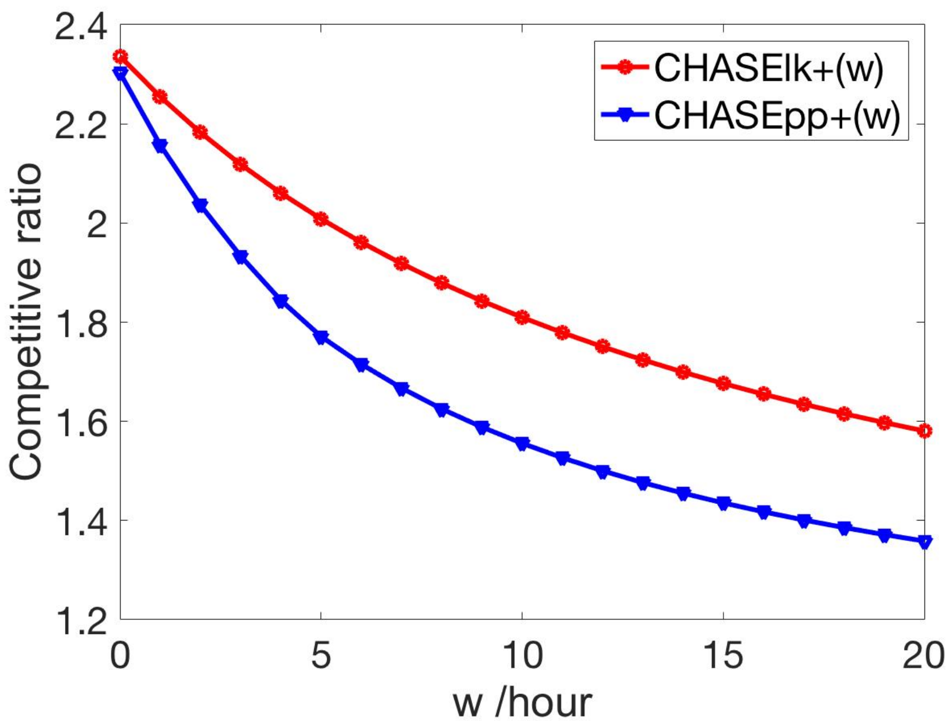

Figure 7.1: Competitive ratio of two algorithms for different prediction window sizes.

7.2 Theoretical Ratio

In this section, we aim to investigate the behavior of the competitive ratio of our proposed online algorithm. To visualize how

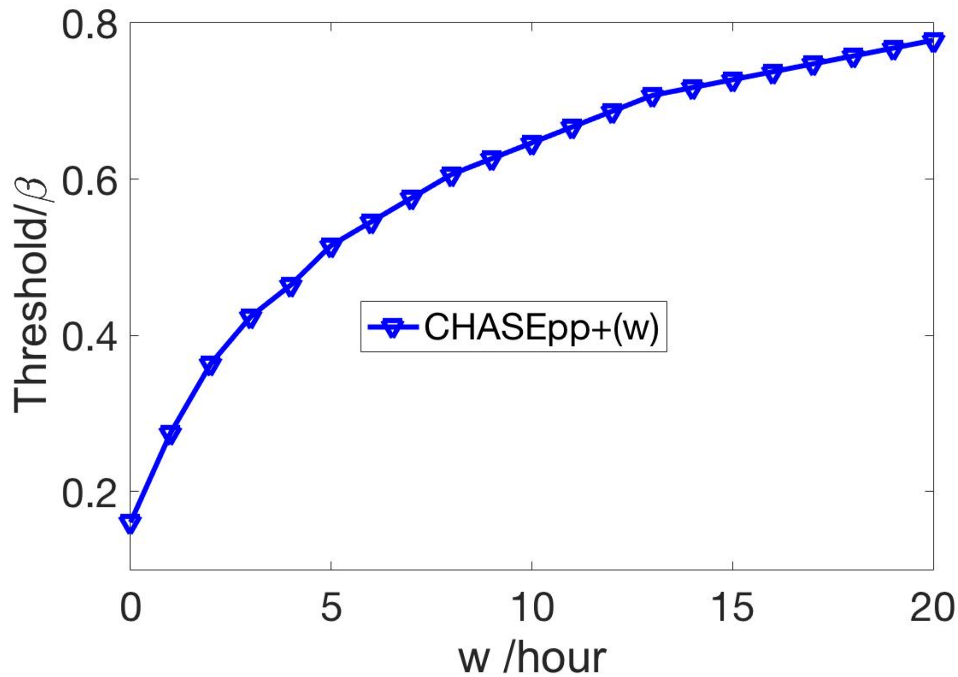

Figure 7.2: The value of the threshold over the startup cost for different window sizes.

the competitive ratio varies as prediction window size changes, we plot the competitive ratio for different values of w in Fig. 7.1. It can be seen that our proposed algorithm always has a better competitive ratio compared to the previous algorithm.

In Fig. 7.2 and Fig. 7.3 the value of the threshold over the startup cost (a^*/β) is shown for different window sizes and different values of α, respectively. By increasing the window size, the threshold monotonically increases and approaches the startup cost β. It can also be seen that by increasing the value of α from 0 to 1, the threshold decreases and reaches zero.

In Fig. 7.4, the competitive ratio value is depicted as a function of the prediction window size (w) and the system parame-

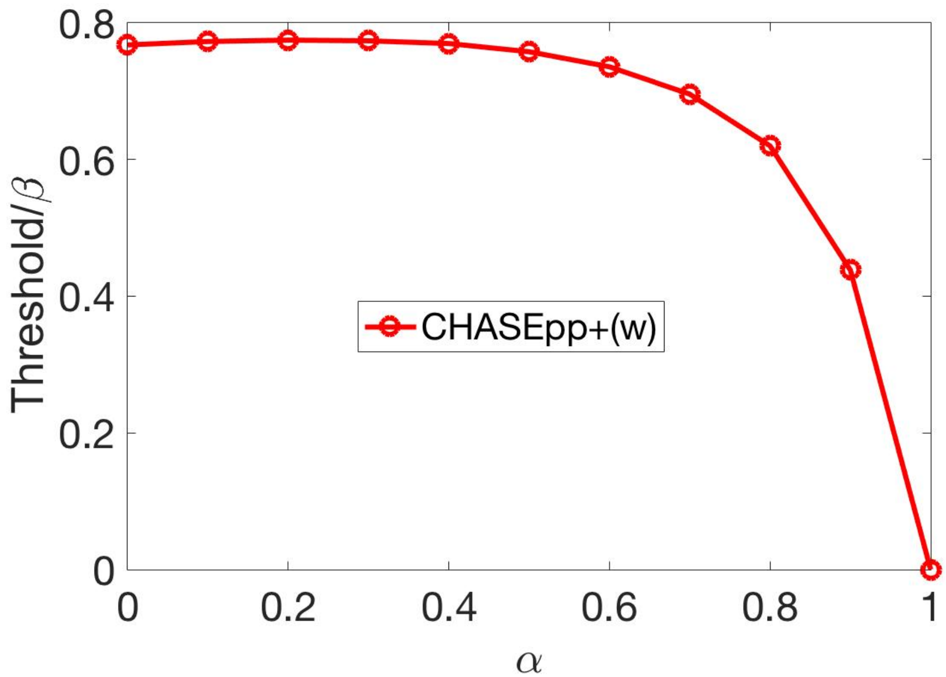

Figure 7.3: The value of the threshold over the startup cost for different values of α.

ters (α). We observe that by increasing the window size from 0 to 20 the competitive ratio decreases and approaches to 1 monotonically.

Recall that α captures the price discrepancy between using the generator and external sources. When α is large it means the economic advantage of using local generation over external sources is small and hence, both online and offline algorithms tend to use local generation less. This makes the online algorithm more competitive to the offline and hence by increasing α from 0 to 1, the competitive ratio decrease from its maximum possible value 3 to 1 monotonically. Note that competitive ratio depends on both prediction window size and the physical pa-

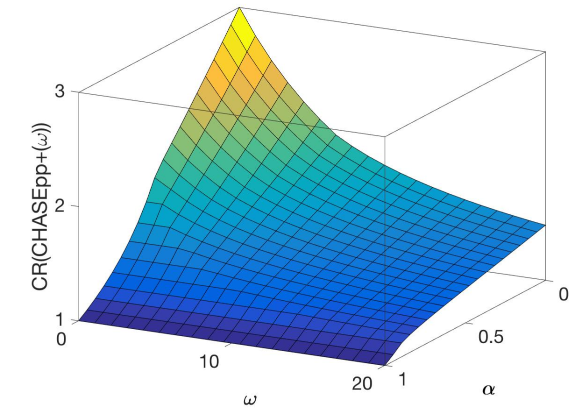

Figure 7.4: Competitive ratio of our proposed online algorithm $\mathsf{CHASEpp}^{+}(w)$ as a function of α and w.

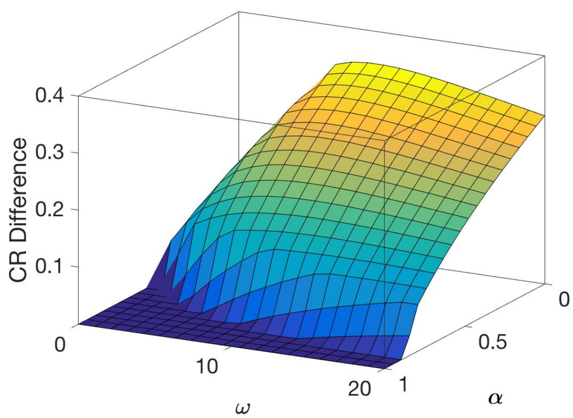

Figure 7.5: Difference between competitive ratio of two algorithms as a function of α and w.

rameters of the system. This means for a fixed window size by changing the system parameters competitive ratio also changes. The importance of this figure is that it can tell us for a particular system with fixed physical parameters how the competitive ratio improves as the prediction window size increases.

In similar figures, the absolute difference between two competitive ratios and the percentage of the competitive ratio im-

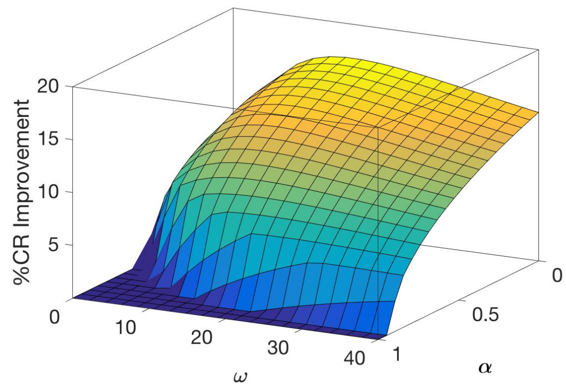

Figure 7.6: Percentage of the improvement of the competitive ratio as a function of α and w.

provement of our algorithm $\mathsf{CHASEpp}^+(w)$ over $\mathsf{CHASElk}^+(w)$ is shown in Fig. 7.5 and Fig. 7.6, respectively. As expected by decreasing the value of α, the competitive ratio improvement increases. The reason is that when α starts to decrease the economic advantage of using local generators over the external grid becomes more prominent and both the offline algorithm and online algorithms tend to use more local generators. However, using the local generation incurs high risk since we have to pay the startup cost to turn on the generator without knowing whether there are sufficient demands to serve in the future. Lacking future knowledge leads to a large performance discrepancy between the $\mathsf{CHASElk}^+(w)$ and the offline optimal solution, making $\mathsf{CHASElk}^+(w)$ less competitive.

By using our new algorithm and checking for sufficiency of the demand in the future window before spending the large start up cost we could decrease the online cost significantly and make $\mathsf{CHASEpp}^+(w)$ more competitive to the offline algorithm. That is why the performance gap between our algorithm and the previous algorithm increases as α decreases.

On the other hand, by increasing the window size the competitive ratio improvement first increases. The reason is that by increasing the window size the effect of the demand density inside the window also increases and by taking this value into account the new algorithm becomes more competitive compared to the previous algorithm. But if we keep increasing the window size, at some point the competitive ratio gap starts to decrease and both competitive ratios tend to reach 1 when the window size goes to infinity. This change of behavior is because even for a long window if the cost difference between using or not using the generator is larger than β, the previous algorithm also turns on the generator in the beginning of the type-1 segment and therefore the new algorithm behaves similar to the previous algorithm.

7.3 The Effect of Prediction Window

In this section, we aim to compare the cost reduction of different algorithms, as the size of the lookahead window increases. Toward this, we change the window size from 0 to 15 and we show the results in Fig. 7.7.

We observe that when the window size is large all the algorithms perform very well and approach the optimal offline solution. On the other hand, when the window size is small RHC performs very poorly while our online algorithm $\mathsf{CHASEpp}^{+}(w)$

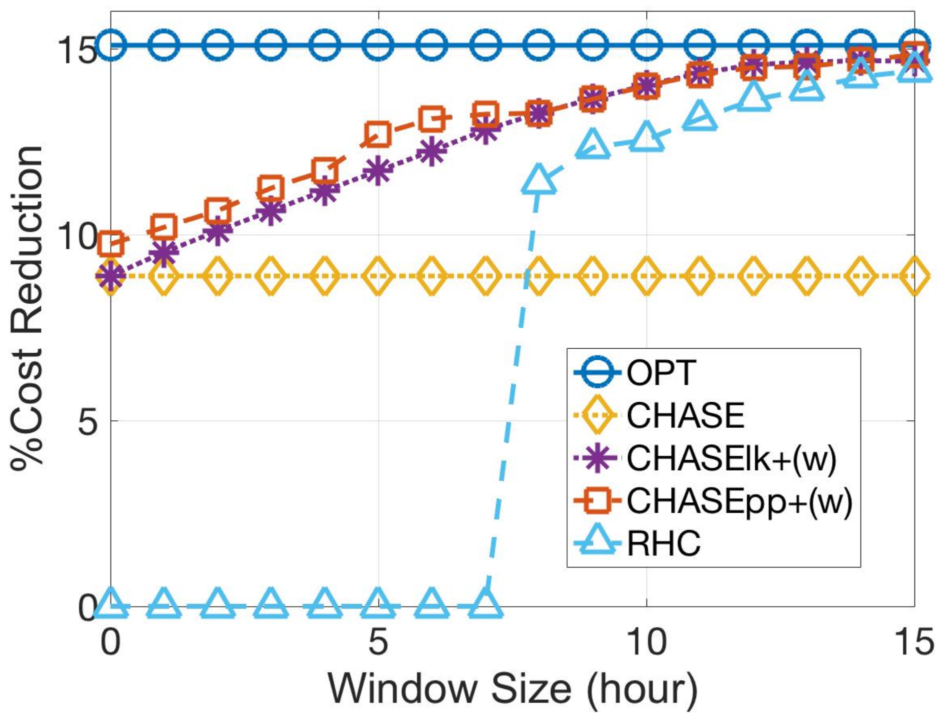

Figure 7.7: Cost reduction of different algorithms as a function of prediction window size.

performs better than the previous algorithm.

Although this improvement is not substantial it is important to note that the improvement largely depends on the input structure and in the following section, we explain when our algorithm can do significantly better than the previous algorithm.

The effect of the demand density: To better understand how the algorithm performance depends on the input and when our algorithm outperforms the previous one we build two inputs that in the first input, two algorithms perform the same where in the second input the new algorithm performs significantly better than the previous one.

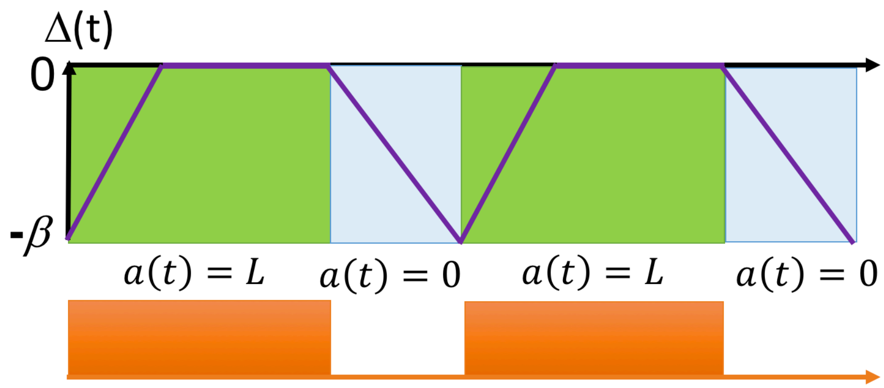

Figure 7.8: Input with $a(t) \in \{0, L\}$, where the cost difference in the window is large.

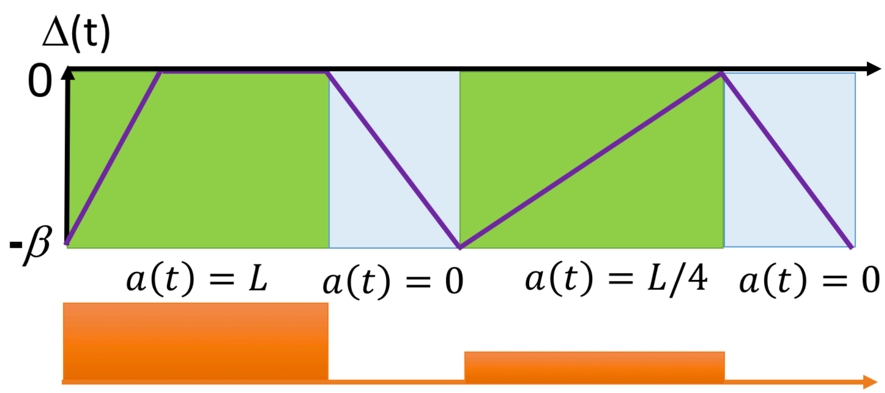

Figure 7.9: Input with $a(t) \in \{0, L/4, L\}$, where the cost difference in the window is small.

In the first input shown in Fig. 7.8, the net electricity demand in the type-1 and type-2 critical segment is set to be $a(t) = L$, and $a(t) = 0$, respectively. We also set the time length of the type-1 and type-2 critical segment to be 30, and 15 time slots, respectively. We set the heat demand $h(t) = \eta a(t)$, and the grid electricity price $p(t) = p_{\max}$ for all the time slots.

The second input shown in Fig. 7.9 is similar to the first input were the only difference is that for half of the type-1 critical segments we have $a(t) = L/4$, which is a smaller demand compared

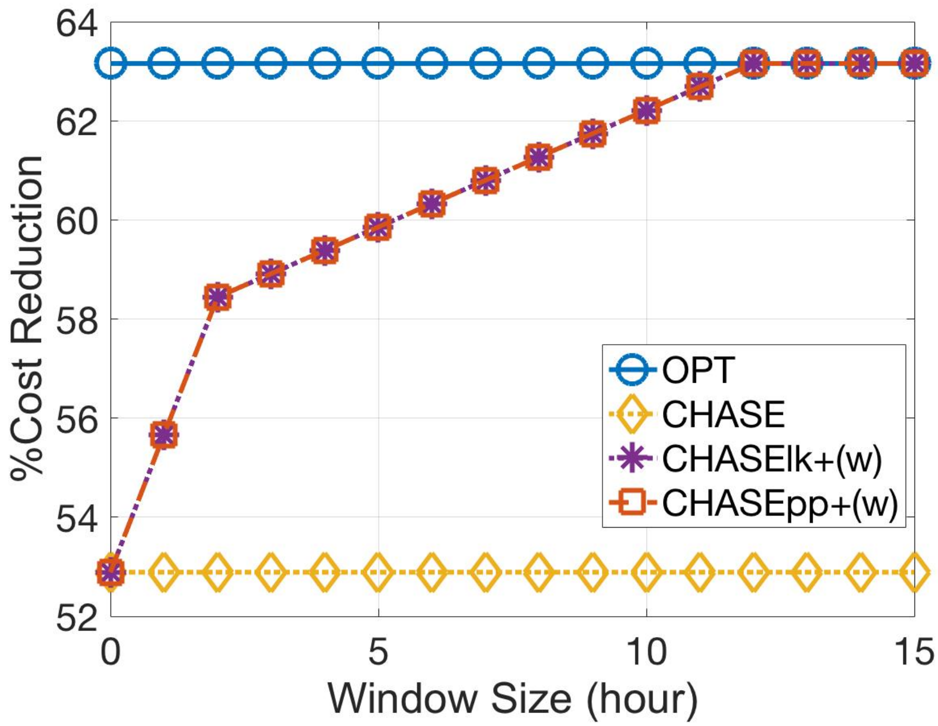

Figure 7.10: The new algorithm $\mathsf{CHASEpp}^+(w)$ performs the same as the previous algorithm $\mathsf{CHASElk}^+(w)$.

to the first input. The cost reduction of the two algorithms for these inputs is depicted in Fig. 7.10 and Fig. 7.11. We observe that in the first input with $a(t) = L$, when $\Delta(t + w)$ reaches zero the cost difference in the window is large and hence both algorithms turn on the generator and they have the same performance. On the other hand, in the second input with $a(t) = L/4$, when $\Delta(t + w)$ reaches zero the cost difference in the window is small and the new algorithms does not turn on the generator and will not spend the additional startup cost, which leads to its better performance.

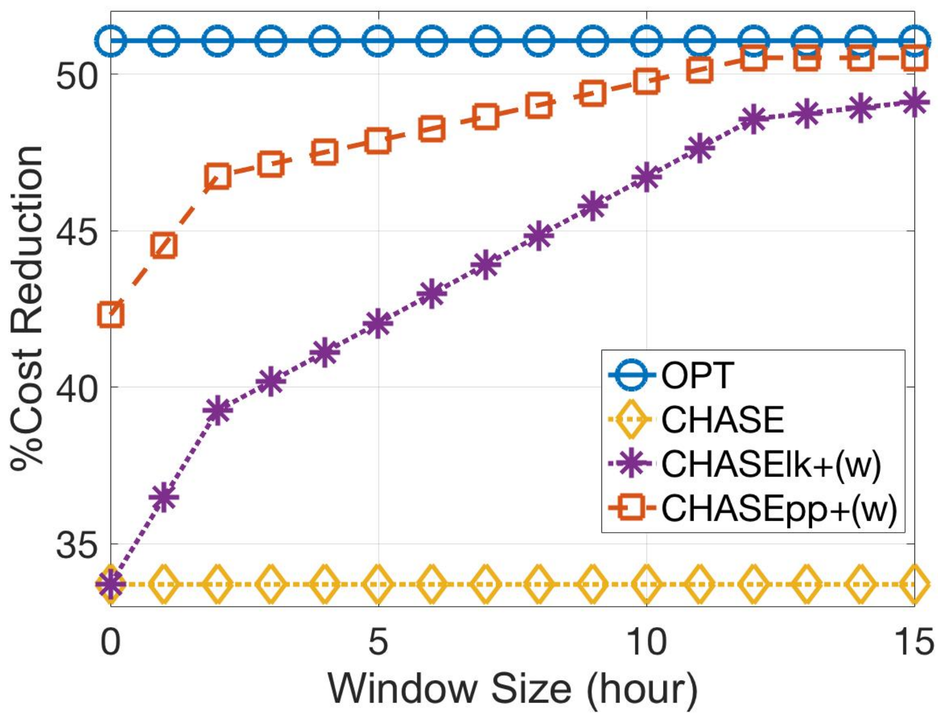

Figure 7.11: The new algorithm $\mathsf{CHASEpp}^{+}(w)$ performs better than previous algorithm $\mathsf{CHASElk}^{+}(w)$.

7.4 The Effect of Prediction Error

All prediction-aware online algorithms are built upon the accurate prediction, which includes the electricity and heat demand, the wind station power output, and the central grid electricity price. By using advanced prediction methods some of this information can be obtained with high accuracy, while others are more likely to come with errors. For example, the day-ahead electricity demand prediction has an error range of $2-3\%$, while the highly fluctuating nature of the wind power makes the next hour's prediction error to usually be around $20-50\%$ [22].

Therefore, it is important to see how the prediction error can

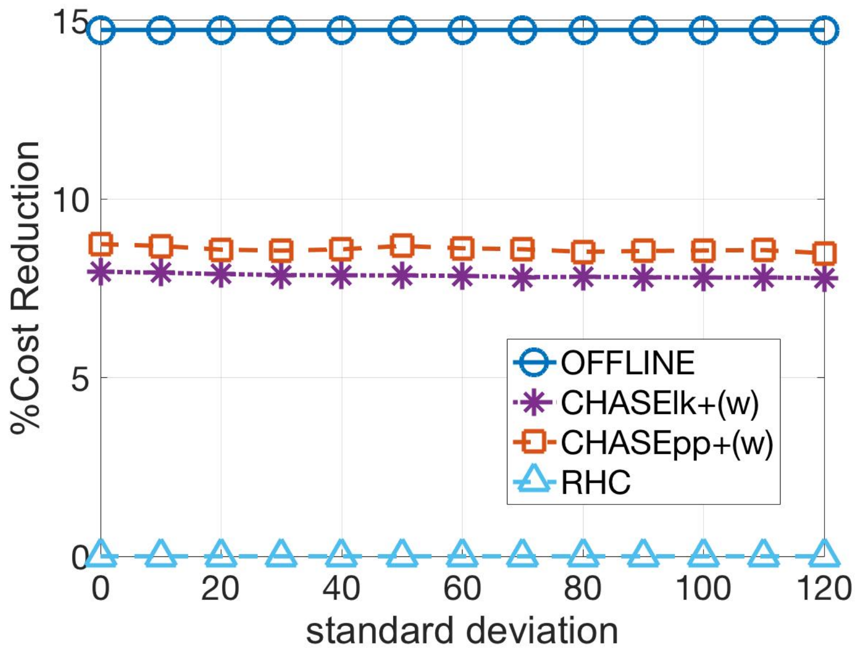

Figure 7.12: Cost reduction as a function of the prediction error size (window size ($w = 1$))

affect the performance of the different algorithms. For the wind power error, we vary the standard deviation of the Gaussian prediction error from 0 to 120% of the total installed capacity, and for the heat demand error, we vary the standard deviation from 0 to 120% of the total peak demand. We report the cost reduction of the algorithms for two different lookahead window sizes of 1 and 3 hours. It is important to note that for a 3-hour prediction window size the errors are often in $20 - 50\%$ range [22]. Therefore, by increasing the standard deviation up to 120%, we are stress-testing the algorithm.

Since the prediction error is randomly generated, we report the average cost reduction across 100 runs and report its value in

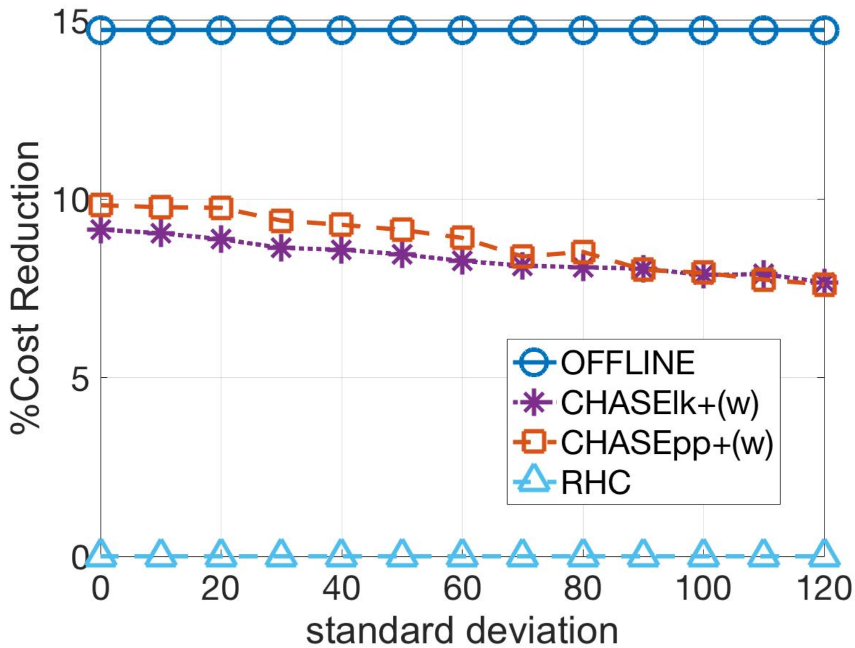

Figure 7.13: Cost reduction as a function of the prediction error size (window size ($w = 3$))

Fig. 7.12 and 7.13. When the window size is small the algorithm performance is very robust to the prediction error, but when the window size increase, both algorithms become more sensitive to the prediction error and we observe that our online algorithm performance approaches to the previous algorithm as prediction error increases.

In Table 7.1 we fix the size of the prediction error and we compare the performance of the algorithm $\mathsf{CHASEpp}^{+}(w)$ for two different window sizes. When we have a more accurate prediction ($sd = 20\%$), by increasing the prediction window size from 1 to 3 the cost reduction increases, which means that if we can keep the prediction error low the larger window size the

Cost reduction	$sd = 20\%$	$sd = 80\%$
$w = 1$	8.58	8.52
$w = 3$	9.76	8.52

Table 7.1: Cost reduction of the algorithm $\mathsf{CHASEpp}^{+}(w)$ for different window sizes and prediction errors.

better performance of our algorithm.

On the hand, when the prediction error is large ($sd = 80\%$), we can see that by increasing the window size value of the cost reduction stays the same, which means that if we have a very noisy prediction, increasing the prediction window size might not be helpful and it might even worsen performance of the algorithm. This is because for a large window size the prediction error of each time slot aggregates over the window and if the window size becomes too large a noisy prediction can be harmful rather than helpful.

□ **End of chapter.**

Chapter 8

Conclusions and Future Works

We study online algorithm design for the microgrid energy generation scheduling problem. The goal is to leverage the accurate prediction of the near future data in the online algorithm design, and investigate its potential benefit both theoretically and practically. We define a new parameter to capture the input behaviour in the prediction window, and by using this parameter and the insight from the previous algorithm we propose a new online algorithm that can improve the algorithm performance. Our competitive ratio analysis shows that our algorithm is always more competitive as compared to the previous algorithm. Our empirical evaluations also demonstrate that our online algorithm performs very well, and it outperforms other known algorithms.

A compelling future direction is to find the best deterministic prediction-aware online algorithm with the minimum competitive ratio. Another interesting future work is to extend our idea to design deterministic online algorithms with performance guarantee that can utilize the widely-available interval prediction of the future demand instead of the accurate prediction.

□ **End of chapter.**

Chapter 9

Appendix

9.1 Proof of Lemma 2

Proof. Without loss of generality let us consider that in a type 1 critical segment the algorithm turns on the generator at time $t = \tilde{T}_i^c + kw - \theta$, where $k \in [0, \infty)$ and $\theta \in [1, w]$ where $k = 0$ and $\theta = w$ gives us the case that we turn on the generator at $t = \tilde{T}_i^c - w$. Here we first consider the case with $k = 0$, and then we extend the result to the general k. In the case with $k = 0$ we turn on the generator at time $t = \tilde{T}_i^c - \theta$ which means we keep the generator off from $t = \tilde{T}_i^c - w$ to $t = \tilde{T}_i^c - \theta$ where $\theta \in [1, w]$.

We denote the outcome of $\mathsf{CHASEpp}(w)$ by $\big(y_{\mathsf{CHASE}(w)}(t)\big)_{t=1}^{T}$, and the outcome of the optimal offline algorithm by $\big(y_{OFA}(t)\big)_{t=1}^{T}$. Let us define $\mathcal{K}_j$ as the set of indices of type-j critical segments i.e.,
$\mathcal{K}_j \triangleq \{\, i \,|\, [T_i^c{+}1, T_{i+1}^c] \textit{is a type-j critical segment in } [0, T]\}$. De-

note the sub-cost for type-j by

$$\begin{aligned}\mathrm{Cost}^{\mathrm{ty}\text{-}j}(y) \;\triangleq\; & \sum_{i\in\mathcal{K}_j}\sum_{t=T_i^c+1}^{T_{i+1}^c} \psi\big(\sigma(t), y(t)\big) \\ & +\beta\cdot[y(t)-y(t-1)]^+ \end{aligned} \tag{9.1}$$

Hence, $\mathrm{Cost}(y)=\sum_{j=0}^{3}\mathrm{Cost}^{\mathrm{ty}\text{-}j}(y)$. We prove by comparing the sub-cost for each type-j.

(**Type-0**): Note that both $y_{\mathrm{OFA}}(t)=y_{\mathrm{CHASE(w)}}(t)=0$ for all $t\in[1,T]$. Hence,

$$\mathrm{Cost}^{\mathrm{ty}\text{-}0}(y_{\mathrm{OFA}})=\mathrm{Cost}^{\mathrm{ty}\text{-}0}(y_{\mathrm{CHASE(w)}}) \tag{9.2}$$

(**Type-1**): Based on the definition of critical segment, we recall that there is an auxiliary point $\tilde{T}_i^c$, such that either $\big(\Delta(T_i^c)=0$ and $\Delta(\tilde{T}_i^c)=-\beta\big)$ or $\big(\Delta(T_i^c)=-\beta$ and $\Delta(\tilde{T}_i^c)=0\big)$. We assumed we turn on the generator at $t=\tilde{T}_i^c-\theta$. Now we focus on the segment $[T_i^c+1,T_{i+1}^c]$. We observe

$$y_{\mathsf{CHASE}(w)}(t)=\begin{cases}0, & \text{for all } t\in[T_i^c+1,\tilde{T}_i^c-\theta),\\ 1, & \text{for all } t\in[\tilde{T}_i^c-\theta,T_{i+1}^c],\end{cases} \tag{9.3}$$

We consider a particular type-1 critical segment: $[T_i^c+1,T_{i+1}^c]$. Note that by the definition of type-1, $y_{\mathrm{OFA}}(T_i^c)=y_{\mathsf{CHASE}(w)}(T_i^c)=0$. $y_{\mathrm{OFA}}(t)$ switches from 0 to 1 at time T_i^c+1, while $y_{\mathsf{CHASE}(w)}$ switches at time $\tilde{T}_i^c-\theta-1$, both incurring startup cost β. The cost difference between $y_{\mathsf{CHASE}(w)}$ and y_{OFA} within

$[T_i^c+1, T_{i+1}^c]$ is

$$\sum_{t=T_i^c+1}^{\tilde{T}_i^c-1} \Big(\psi\Big(\sigma(t),0\Big) - \psi\Big(\sigma(t),1\Big)\Big) + \beta - \beta \tag{9.4}$$
$$= \sum_{t=T_i^c+1}^{\tilde{T}_i^c-\theta-1} \delta(t) = \Delta(\tilde{T}_i^c-\theta-1) - \Delta(T_i^c) = -q_i^1 + \beta$$

where $q_i^1 \triangleq -\Delta(\tilde{T}_i^c-\theta-1)$.

Let the number of type-j critical segments be $m_j \triangleq |\mathcal{K}_j|$.

$$\mathrm{Cost}^{\text{ty-1}}(y_{\mathsf{CHASE}(w)}) \le \mathrm{Cost}^{\text{ty-1}}(y_{\mathrm{OFA}}) + m_1\cdot\beta - \sum_{i\in\mathcal{K}_1} q_i^1 \tag{9.5}$$

(**type-2**) and (**type-3**): We derive similarly for $j=2$ or 3 as

$$\mathrm{Cost}^{\text{ty-}j}(y_{\mathsf{CHASE}(w)}) \le \mathrm{Cost}^{\text{ty-}j}(y_{\mathrm{OFA}}) + m_j\cdot\beta - \sum_{i\in\mathcal{K}_j} q_i^j. \tag{9.6}$$

where $q_i^j \triangleq \beta + \Delta(\tilde{T}_i^c - w - 1)$.

Note that $|q_i^j| \le \beta$ for all i,j. Furthermore, we note $m_1 = m_2+m_3$, because it takes equal numbers of critical segments for increasing $\Delta(\cdot)$ from $-\beta$ to 0 and for decreasing from 0 to $-\beta$. We obtain

$$\frac{\mathrm{Cost}(y_{\mathsf{CHASE}(w)})}{\mathrm{Cost}(y_{\mathrm{OFA}})} = \frac{\sum_{j=0}^3 \mathrm{Cost}^{\text{ty-}j}(y_{\mathsf{CHASE}(w)})}{\sum_{j=0}^3 \mathrm{Cost}^{\text{ty-}j}(y_{\mathrm{OFA}})}$$
$$\le 1 + \frac{2m_1\beta + \sum_{i\in\mathcal{K}_1} q_i^1 - \sum_{i\in\mathcal{K}_2} q_i^2 - \sum_{i\in\mathcal{K}_3} q_i^3}{\sum_{j=0}^3 \mathrm{Cost}^{\text{ty-}j}(y_{\mathrm{OFA}})} \tag{9.7}$$

It should be noted that in the calculation a type-3 critical segment is exactly the same as a type-2 critical segment and hence in the rest of the calculation we just consider type-2 critical segments. As a result we can write $m_1 = m_2$ for the ease of calculation. We have

$$\begin{aligned}\frac{\mathrm{Cost}(y_{\mathsf{CHASE}(w)})}{\mathrm{Cost}(y_{\mathrm{OFA}})} &\leq 1 + \frac{2m_1\beta - \sum_{i\in\mathcal{K}_1} q_i^1 - \sum_{i\in\mathcal{K}_2} q_i^2}{\sum_{j=0}^{2} \mathrm{Cost}^{\mathrm{ty}\text{-}j}(y_{\mathrm{OFA}})} \\ &\leq 1 + \begin{cases} 0 & \text{if } m_1 = 0, \\ \dfrac{2m_1\beta - \sum_{i\in\mathcal{K}_1} q_i^1 - \sum_{i\in\mathcal{K}_2} q_i^2}{\sum_{j=0}^{2} \mathrm{Cost}^{\mathrm{ty}\text{-}j}(y_{\mathrm{OFA}})} & \text{otherwise} \end{cases} \end{aligned} \tag{9.8}$$

By Lemma 4, and Lemma 5 and simplifications, we obtain:

$$\begin{aligned}\frac{\mathrm{Cost}(y_{\mathsf{CHASE}(w)})}{\mathrm{Cost}(y_{\mathrm{OFA}})} &\leq 1 + \Big(1 - \frac{Lc_{\mathrm{o}} + c_{\mathrm{m}}}{L(p_{\max} + \eta \cdot c_{\mathrm{g}})}\Big) \cdot \max_{q\in\{0, wc_{\mathrm{m}}\}} \\ &\Big\{\frac{(2\beta - q)}{\beta + \big(2wc_{\mathrm{m}} - q + \frac{c_{\mathrm{o}}}{p_{\max}+\eta\cdot c_{\mathrm{g}}} a\big)\big(1 - \frac{c_{\mathrm{m}}}{L(p_{\max}+\eta\cdot c_{\mathrm{g}} - c_{\mathrm{o}})}\big)}\Big\}\end{aligned} \tag{9.9}$$

We denote this value as $R_{\mathrm{on}}(a)$. This is the performance ratio for the case with $k = 0$. Now if we increase k, by using the same process we can see that both online and offline cost increase. Using the result from Lemma 3, if we change to general k, the increment ratio is as follows:

$$\frac{\text{online cost increment}}{\text{offline cost increment}} \leq \frac{k\big((w+1)c_{\mathrm{m}} + a\big)}{k\big((w+1)c_{\mathrm{m}} + \frac{c_{\mathrm{o}}}{p_{\max}+\eta\cdot c_{\mathrm{g}}} a\big)} \tag{9.10}$$

We denote this ratio as $R_{\mathrm{off}}(a)$. This is the ratio of the increment of the numerator and denominator in the $R_{\text{on}}(a)$. If this ratio is larger than the previous one $R_{\text{off}}(a) > R_{\text{on}}(a)$, then by increasing k, the value of the performance ratio keeps increasing and when k goes to ∞ this competitive ratio goes to $R_{\text{off}}(a)$. On the other hand, if $R_{\text{off}}(a) \leq R_{\text{on}}(a)$, by increasing the value of k, value of the competitive ratio will not increase and is still upper bounded by $R_{\text{on}}(a)$. This shows that the competitive ratio is upper bonded by the maximum of $R_{\text{on}}(a)$, and $R_{\text{off}}(a)$. In Lemma 6 we show that $R_{\text{off}}(a)$ is always an increasing function while $R_{\text{on}}(a)$ is always a decreasing function. This completes the proof. □

Lemma 4. *For the (**type-1**), we have*

$$\begin{aligned} & \text{Cost}^{\text{ty-1}}(y_{\text{OFA}}) \\ \geq\ & m_1\beta + \sum_{i\in\mathcal{K}_1}\Big(\frac{(q_i^1+\beta)(Lc_{\text{o}}+c_{\text{m}})}{L\big(p_{\max}+\eta\cdot c_{\text{g}}-c_{\text{o}}\big)-c_{\text{m}}} \\ & \frac{p_{\max}+\eta\cdot c_{\text{g}}}{p_{\max}+\eta\cdot c_{\text{g}}-c_{\text{o}}}\big(wc_{\text{m}}+\frac{c_{\text{o}}}{p_{\max}+\eta\cdot c_{\text{g}}}a\big)\Big). \end{aligned} \tag{9.11}$$

Proof. Consider a particular type-1 segment $[T_i^c+1, T_{i+1}^c]$. For the offline cost, we have:

$$\begin{aligned} \text{Cost}^{\text{ty-1}}(y_{\text{OFA}}) &= \beta + \sum_{t=T_i^c+1}^{T_{i+1}^c}\psi\big(\sigma(t),1\big) \\ &= \beta + (T_{i+1}^c - T_i^c)c_{\text{m}} + \sum_{t=T_i^c+1}^{T_{i+1}^c}\Big(\psi\big(\sigma(t),1\big)-c_{\text{m}}\Big). \end{aligned} \tag{9.12}$$

By [26, Lemma. 4] and simplification we obtain

$$\begin{aligned}
&\mathrm{Cost}^{\mathrm{up}} \geq \beta + (T^c_{i+1} - T^c_i)c_{\mathrm{m}} + \qquad (9.13)\\
&\frac{c_{\mathrm{o}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}}\Big(\sum_{t=T^c_i+1}^{T^c_{i+1}} \delta(t) + (T^c_{i+1} - T^c_i)c_{\mathrm{m}}\Big)\\
&= \beta + \frac{p_{\max} + \eta \cdot c_{\mathrm{g}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}}(T^c_{i+1} - T^c_i)c_{\mathrm{m}} \qquad (9.14)\\
&+ \frac{c_{\mathrm{o}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}}\sum_{t=T^c_i+1}^{T^c_{i+1}} \delta(t)
\end{aligned}$$

Now we need to find the lower bound of both $T^c_{i+1} - T^c_i$ and $\sum_{t=T^c_i+1}^{T^c_{i+1}} \delta(t)$ in the following two steps.

Step 1: We write the lower bound of $\sum_{t=T^c_i+1}^{T^c_{i+1}} \delta(t)$ as follows:

$$\begin{aligned}
\sum_{t=T^c_i+1}^{T^c_{i+1}} \delta(t) &= \sum_{t=T^c_i+1}^{\tilde{T}^c_i-\theta-1} \delta(t) + \sum_{\tilde{T}^c_i-\theta}^{T^c_{i+1}} \delta(t)\\
&= \Delta(\tilde{T}^c_i - \theta - 1) - \Delta(T^c_i) + \sum_{\tilde{T}^c_i-\theta}^{T^c_{i+1}} \delta(t)\\
&= \beta - q^1_i + \sum_{\tilde{T}^c_i-\theta}^{T^c_{i+1}} \delta(t) \geq \beta - q^1_i + a \qquad (9.15)
\end{aligned}$$

Step 2: To find the lower bound of the length of the interval $[T^c_i{+}1, T^c_{i+1}]$, we have two cases with $\theta = w$ or $\theta < w$ we calculate the lower bound as follows:

Case 1: If $\theta = w$, we can see that $[T^c_i + 1, T^c_{i+1}]$ has two part as $[T^c_i + 1, \tilde{T}^c_i - w - 1]$ and $[\tilde{T}^c_i - w, T^c_{i+1}]$. We note that $\big(\tilde{T}^c_i - w - 1 - T^c_i\big)$ is lower bounded by the steepest descend when

$p(t) = p_{\max}$, $a(t) = L$ and $h(t) = \eta L$,

$$\tilde{T}_i^c - w - 1 - T_i^c \geq \frac{\beta - q_i^1}{L\big(p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}\big) - c_{\mathrm{m}}} \tag{9.16}$$

and for the second part we have

$$T_{i+1}^c - \tilde{T}_i^c + w + 1 \geq w \tag{9.17}$$

which means

$$T_{i+1}^c - T_i^c \geq \frac{\beta - q_i^1}{L\big(p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}\big) - c_{\mathrm{m}}} + w \tag{9.18}$$

Case 2: On the other hand, when $\theta < w$, we know the length of the interval $[\tilde{T}_i^c - w, \tilde{T}_i^c]$ is $w+1$ time slot and its cost difference is less than a, hence to calculate the total $[T_i^c + 1, T_{i+1}^c]$ length we have

$$\begin{aligned}
&\sum_{t=T_i^c+1}^{T_{i+1}^c} \delta(t) \geq \beta - q_i^1 + a \\
&\sum_{t=T_i^c+1}^{T_{i+1}^c} \delta(t) - \sum_{\tilde{T}_i^c - w}^{\tilde{T}_i^c} \delta(t) + \sum_{\tilde{T}_i^c - w}^{\tilde{T}_i^c} \delta(t) \geq \beta - q_i^1 + a \\
&\sum_{t=T_i^c+1}^{T_{i+1}^c} \delta(t) - \sum_{\tilde{T}_i^c - w}^{\tilde{T}_i^c} \delta(t) \geq \beta - q_i^1
\end{aligned} \tag{9.19}$$

where the last inequality comes from the fact that $\sum_{\tilde{T}_i^c - w}^{\tilde{T}_i^c} \delta(t) \leq a$. We note that $\big(T_{i+1}^c - T_i^c - (w+1)\big)$ is lower bounded by the steepest descend when $p(t) = p_{\max}$, $a(t) = L$ and $h(t) = \eta L$,

$$T_{i+1}^c - T_i^c - (w+1) \geq \frac{\beta - q_i^1}{L\big(p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}\big) - c_{\mathrm{m}}}$$

$$T_{i+1}^c - T_i^c \geq \frac{\beta - q_i^1}{L\big(p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}\big) - c_{\mathrm{m}}} + w \tag{9.20}$$

So one can see that in both of these cases we always have

$$T_{i+1}^c - T_i^c \geq \frac{\beta - q_i^1}{L\big(p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}\big) - c_{\mathrm{m}}} + w \tag{9.21}$$

length By Eqns. (9.15)-(9.21), we obtain

$$\begin{aligned}
&\mathrm{Cost}^{\mathrm{up}} \geq \beta + \\
&\frac{p_{\max} + \eta \cdot c_{\mathrm{g}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}} \Big(\frac{\beta - q_i^1}{L(p_{\max} + \eta c_{\mathrm{g}} - c_{\mathrm{o}} - \frac{c_{\mathrm{m}}}{L})} + w \Big) c_{\mathrm{m}} \\
&+ \frac{c_{\mathrm{o}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}} \big(\beta - q_i^1 + a\big) \\
&= \beta + \frac{(\beta - q_i^1)(Lc_{\mathrm{o}} + c_{\mathrm{m}})}{L(p_{\max} + \eta c_{\mathrm{g}} - c_{\mathrm{o}} - \frac{c_{\mathrm{m}}}{L})} + \\
&\frac{p_{\max} + \eta \cdot c_{\mathrm{g}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}} \big(wc_{\mathrm{m}} + \frac{c_{\mathrm{o}}}{p_{\max} + \eta \cdot c_{\mathrm{g}}} a\big)
\end{aligned} \tag{9.22}$$

Since there are m_1 type-1 critical segments, according to Eqna. (9.22), we obtain

$$\begin{aligned}
&\mathrm{Cost}^{\text{ty-1}}(y_{\mathrm{OFA}}) \\
\geq\ & m_1\beta + \sum_{i \in \mathcal{K}_1} \Big(\frac{(\beta - q_i^1)(Lc_{\mathrm{o}} + c_{\mathrm{m}})}{L\big(p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}\big) - c_{\mathrm{m}}} \\
& \frac{p_{\max} + \eta \cdot c_{\mathrm{g}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}} \big(wc_{\mathrm{m}} + \frac{c_{\mathrm{o}}}{p_{\max} + \eta \cdot c_{\mathrm{g}}} a\big) \Big).
\end{aligned} \tag{9.23}$$

□

Lemma 5. *For (**type-2**), we have*

$$\mathrm{Cost}^{\text{ty-2}}(y_{\mathrm{OFA}}) \geq \sum_{i \in \mathcal{K}_2} \Big(\frac{p_{\max} + \eta \cdot c_{\mathrm{g}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}} (w \cdot c_{\mathrm{m}} - q_i^2) \Big) \tag{9.24}$$

Proof. Consider a particular type-2 segment $[T_i^c + 1, T_{i+1}^c]$. We bound $\mathrm{Cost}^{\text{ty-2}}$ as follows:

$$\mathrm{Cost}^{\text{ty-2}} = \sum_{t=T_i^c+1}^{T_{i+1}^c} \psi\big(\sigma(t), 0\big) \tag{9.25}$$

By [26, Lemma. 4] and simplification we obtain

$$\mathrm{Cost}^{\text{ty-2}} \geq \tag{9.26}$$

$$\frac{p_{\max} + \eta \cdot c_{\mathrm{g}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}} \Bigg(\sum_{t=T_i^c+1}^{T_{i+1}^c} \delta(t) + (T_{i+1}^c - T_i^c) c_{\mathrm{m}} \Bigg)$$

$$= \frac{p_{\max} + \eta \cdot c_{\mathrm{g}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}} \Bigg(-\beta + (T_{i+1}^c - T_i^c) c_{\mathrm{m}} \Bigg) \tag{9.27}$$

Furthermore, we note that $\big(T_{i+1}^c - T_i^c\big)$ is lower bounded by the steepest descend when $\min\{a(t), h(t)\} = 0$,

$$T_{i+1}^c - T_i^c \geq w + \frac{\beta - q_i^2}{c_{\mathrm{m}}} \tag{9.28}$$

By Eqns. (9.27)-(9.28), we obtain

$$\mathrm{Cost}^{\text{ty-2}} \geq \frac{p_{\max} + \eta \cdot c_{\mathrm{g}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}} (w \cdot c_{\mathrm{m}} - q_i^2) \tag{9.29}$$

Since there are m_2 type-2 critical segments, according to Eqna. (9.29), we obtain

$$\begin{aligned} & \mathrm{Cost}^{\text{ty-2}}(y_{\mathrm{OFA}}) \\ \geq & \sum_{i \in \mathcal{K}_2} \Big(\frac{p_{\max} + \eta \cdot c_{\mathrm{g}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}} (w \cdot c_{\mathrm{m}} - q_i^2) \Big). \end{aligned} \tag{9.30}$$

□

Lemma 6. *$R_{\mathrm{on}}(a)$ is always a decreasing function of a and $R_{\mathrm{off}}(a)$ is always an increasing function of a.*

Proof. $R_{\mathrm{on}}(a)$: To prove that $R_{\mathrm{on}}(a)$ is always a decreasing function first we take the derivative as a function of a. To compute the derivative we only consider the maximization part of $R_{\mathrm{on}}(a)$. The denominator of the function is always positive and the numerator is given by

$$-\big(\frac{(2\beta - q)c_{\mathrm{o}}}{p_{\max} + \eta \cdot c_{\mathrm{g}}} a\big)\big(1 - \frac{c_{\mathrm{m}}}{L(p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}})}\big). \tag{9.31}$$

Note that we have $q \in \{0, wc_{\mathrm{m}}\}$. If $wc_{\mathrm{m}} < 2\beta$ the derivative is always negative. If $wc_{\mathrm{m}} \geq 2\beta$ we show that we have $q = 0$ in the maximization part of the function which again shows that derivative is negative. To show that for $wc_{\mathrm{m}} \geq 2\beta$ we have $q = 0$, we first take the derivative as a function of q and we can see that the denominator of the function is always positive and the numerator is given by

$$-\beta - \big(2(wc_{\mathrm{m}} - \beta) + \tfrac{c_{\mathrm{o}}}{p_{\max}+\eta \cdot c_{\mathrm{g}}} a\big)\big(1 - \tfrac{c_{\mathrm{m}}}{L(p_{\max}+\eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}})}\big). \tag{9.32}$$

We can see that for $wc_{\mathrm{m}} \geq \beta$ value of the derivative is always negative which means that for $wc_{\mathrm{m}} \geq 2\beta$ the maximum value of $R_{\mathrm{on}}(a)$ happens at $q = 0$. This prove that for both cases the derivative is negative and hence $R_{\mathrm{on}}(a)$ is always a decreasing function.

$R_{\mathrm{off}}(a)$: Now we show that $R_{\mathrm{off}}(a)$ is an increasing function. We take the derivative and we can see that the denominator of the function is always positive and the numerator is given by

$$\frac{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}}{p_{\max} + \eta \cdot c_{\mathrm{g}}}(w+1)c_{\mathrm{m}}, \tag{9.33}$$

which is always positive and hence $R_{\mathrm{off}}(a)$ is always an increasing function. This completes the proof. □

9.2 Proof of Lemma 3

Proof. By [26, Lemma. 4] and simplification we know that in a type-1 critical segment for a window with $\Delta_t^{t+w} = a$, for the offline cost we have

$$\mathrm{Cost}(y_{\mathrm{OFA}}) \geq \frac{p_{\max} + \eta \cdot c_{\mathrm{g}}}{p_{\max} + \eta \cdot c_{\mathrm{g}} - c_{\mathrm{o}}}\left((w+1)c_{\mathrm{m}} + \frac{c_{\mathrm{o}}}{p_{\max} + \eta \cdot c_{\mathrm{g}}}a\right) \tag{9.34}$$

On the other hand, if in the type-1 critical segment the online keep the generator off in this window, the cost difference between the online and the offline is equal to a, which means

$$\mathrm{Cost}(y_{\mathsf{CHASEpp}(w)}) - \mathrm{Cost}(y_{\mathrm{OFA}}) = a \tag{9.35}$$

Hence we have

$$\begin{aligned}\frac{\text{Cost}(y_{\mathsf{CHASEpp}(w)})}{\text{Cost}(y_{\text{OFA}})} &= 1 + \frac{a}{\text{Cost}(y_{\text{OFA}})} \\ &\leq \frac{(w+1)c_{\text{m}} + a}{(w+1)c_{\text{m}} + \frac{c_{\text{o}}}{p_{\max}+\eta\cdot c_{\text{g}}}a}\end{aligned} \tag{9.36}$$

This completes the proof. □

9.3 Proof of Theorem 4

Proof. From Lemma 2 we have

$$\mathsf{CR}(\mathcal{A}(a^*)) = \max\{R_{\text{on}}(a^*), R_{\text{off}}(a^*)\} \tag{9.37}$$

and from the definition of the optimal threshold a^* in (6.8d) we have

$$\max\{R_{\text{on}}(a^*), R_{\text{off}}(a^*)\} = R_{\text{on}}(a^*) \tag{9.38}$$

Therefore for the competitive ratio we have:

$$\mathsf{CR}(\mathcal{A}(a^*)) = R_{\text{on}}(a^*). \tag{9.39}$$

By using the definition of α in 4.8 and simplification we obtain the result which completes the proof. □

□ **End of chapter.**

Bibliography

[1] California commercial end-use survey. Internet:http://capabilities.itron.com/CeusWeb/.

[2] European commission, 2030 climate and energy framework. Internet:https://ec.europa.eu/clima/policies/strategies.

[3] Green energy. Internet:http://www.green-energy-uk.com/whatischp.html.

[4] National renewable energy laboratory. Internet:http://wind.nrel.gov.

[5] Navigant research, microgrid deployment tracker. Internet:https://www.navigantresearch.com.

[6] Pacific gas and electric company. Internet:http://www.pge.com/nots/rates/tariffs/rateinfo.shtml.

[7] Statistical office of the european union, renewable energy statistics. Internet:https://ec.europa.eu/eurostat.

[8] Tecogen. Internet:http://www.tecogen.com.

[9] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis.* cambridge university press, 2005.

[10] X. Cao and K. J. R. Liu. On the time-varying constraints and bandit feedback of online convex optimization. *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, 2018.

[11] X. Cao, J. Zhang, and H. V. Poor. A virtual-queue based algorithm for constrained online convex optimization with applications to data center resource allocation. *IEEE Journal on Selected Topics in Signal Processing*, 12(4):703–716, 2018.

[12] M. Carrión and J. Arroyo. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Trans. Power Systems*, 21(3):1371–1378, 2006.

[13] N. Chen, A. Agarwal, A. Wierman, S. Barman, and L. L. Andrew. Online convex optimization using predictions. *SIGMETRICS Perform. Eval. Rev.*, 43(1):191–204, 2015.

[14] N. Chen, J. Comden, Z. Liu, A. Gandhi, and A. Wierman. Using predictions in online optimization: Looking forward with an eye on the past. In *Proc. of ACM SIGMETRICS*, pages 193–206, 2016.

[15] P. Chen and H. Chang. Large-scale economic dispatch by genetic algorithm. *IEEE Trans. Power Systems*, 10(4):1919–1926, 1995.

[16] Z. Gaing. Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE Trans. Power Systems*, 18(3):1187–1195, 2003.

[17] X. Guan, Q. Zhai, and A. Papalexopoulos. Optimization based methods for unit commitment: Lagrangian relaxation versus general mixed integer programming. In *Proc. IEEE PES General Meeting*, 2003.

[18] T. Guo, M. Henwood, and M. van Ooijen. An algorithm for combined heat and power economic dispatch. *IEEE Trans. Power Systems*, 11(4):1778–1784, 1996.

[19] M. H. Hajiesmaili, C. K. Chau, M. Chen, and L. Huang. Online microgrid energy generation scheduling revisited: The benefits of randomization and interval prediction. In *Proc. ACM e-Energy*, 2016.

[20] A. Hawkes and M. Leach. Modelling high level system design and unit commitment for a microgrid. *Applied energy*, 86(7):1253–1265, 2009.

[21] C. Hernandez-Aramburo, T. Green, and N. Mugniot. Fuel consumption minimization of a microgrid. *IEEE Trans. Industry Applications*, 41(3):673–681, 2005.

[22] B. Hodge and M. Milligan. Wind power forecasting error distributions over multiple timescales. In *Proc. IEEE PES General Meeting*, 2011.

[23] S. Kazarlis, A. Bakirtzis, and V. Petridis. A genetic algorithm solution to the unit commitment problem. *IEEE Trans. Power Systems*, 11(1):83–92, 1996.

[24] W. Kwon and A. Pearson. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Trans. Automatic Control*, 22(5):838–842, 1977.

[25] M. Lin, A. Wierman, L. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *Proc. IEEE INFOCOM*, 2011.

[26] L. Lu, J. Tu, C.-K. Chau, M. Chen, and X. Lin. Online energy generation scheduling for microgrids with intermittent energy sources and co-generation. In *Proc. of ACM SIGMETRICS*, pages 53–66, 2013.

[27] T. Lu and M. Chen. Simple and effective dynamic provisioning for power-proportional data centers. In *Proc. CISS*, 2012.

[28] B. Narayanaswamy, V. Garg, and T. Jayram. Online optimization for the smart (micro) grid. In *Proc. ACM International Conference on Future Energy Systems*, 2012.

[29] M. Neely. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*, 3(1):1–211, 2010.

[30] N. Padhy. Unit commitment-a bibliographical survey. *IEEE Trans. Power Systems*, 19(2):1196–1205, 2004.

[31] A. Selvakumar and K. Thanushkodi. A new particle swarm optimization solution to nonconvex economic dispatch problems. *IEEE Trans. Power Systems*, 22(1):42–51, 2007.

[32] W. Snyder, H. Powell, and J. Rayburn. Dynamic programming approach to unit commitment. *IEEE Trans. Power Systems*, 2(2):339–348, 1987.

[33] S. Takriti, J. Birge, and E. Long. A stochastic model for the unit commitment problem. *IEEE Trans. Power Systems*, 11(3):1497–1508, 1996.

[34] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman. Dynamic provisioning in next-generation data centers with on-site power production. In *Proceedings of the fourth international conference on Future energy systems*, pages 137–148. ACM, 2013.

[35] R. Urgaonkar, B. Urgaonkar, M. Neely, and A. Sivasubramaniam. Optimal power cost management using stored energy in data centers. In *Proc. ACM SIGMETRICS*, 2011.

[36] Y. Zhang, M. H. Hajiesmaili, and M. Chen. Peak-aware online economic dispatching for microgrids. In *Proceedings of the ACM Sixth International Conference on Future Energy Systems (e-Energy)*, pages 27–36. ACM, 2015.

[37] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. *School of Computer Science, Carnegie Mellon University*, 2003.