Competitive Prediction-Aware Online Algorithms for Energy Generation Scheduling in Microgrids

Ali Menati
School of Data Science
City University of Hong Kong

Sid Chi-Kin Chau
Research School of Computer Science
Australian National University

Minghua Chen*
School of Data Science
City University of Hong Kong

ABSTRACT

Online decision-making in the presence of uncertain future information is abundant in many problem domains. In the critical problem of energy generation scheduling for microgrids, one needs to decide when to switch energy supply between a cheaper local generator with startup cost and the costlier on-demand external grid, considering intermittent renewable generation and fluctuating demands. Without knowledge of future input, competitive online algorithms are appealing as they provide optimality guarantees against the optimal offline solution. In practice, however, future input, e.g., wind generation, is often predictable within a limited time window, and can be exploited to further improve the competitiveness of online algorithms. In this paper, we exploit the structure of information in the prediction window to design a novel prediction-aware online algorithm for energy generation scheduling in microgrids. Our algorithm achieves the best competitive ratio to date for this important problem, which is at most $3 - 2/(1 + O(\frac{1}{w}))$, where w is the prediction window size. We also characterize a non-trivial lower bound of the competitive ratio and show that the competitive ratio of our algorithm is only 9% away from the lower bound, when a few hours of prediction is available. Simulation results based on real-world traces corroborate our theoretical analysis and highlight the advantage of our new prediction-aware design.

CCS CONCEPTS

• **Theory of computation → Online algorithms**.

KEYWORDS

microgrids, prediction-aware online algorithm, energy generation scheduling, competitive analysis

*Corresponding author: Minghua Chen.

1 INTRODUCTION

Central to online decision-making problems is the presence of future information, which, if available, determines the optimal decisions taken currently. Without knowledge of future information, competitive online algorithms are robust decision-making algorithms that can offer a worst-case guarantee to their sub-optimal decisions, against the optimal offline decisions with complete future information. In microgrids, it is essential to serve the fluctuating demands using local generators, intermittent renewable energy sources, and an external grid with time-varying tariffs. This is a well-studied class of problems in smart grid literature (including economic dispatching [14] and unit commitment problems [24]). It is appealing to employ competitive online algorithms for efficient energy management in microgrids [36]. In practice, however, prediction is often plausible within a limited time window. For example, in smart grid, the advances of machine learning and big data analytics enable relatively accurate renewable energy forecasting with several hours ahead [51]. The availability of predicted future information will certainly enhance the design of online decision-making algorithms, providing the missing information for optimal decisions. Nonetheless, prediction is never perfect. When we consider only a limited time window of accurate prediction, it may not be sufficient to determine the current optimal decisions. Thus, a worst-case guarantee is still desirable to benchmark an online algorithm's sub-optimal decisions using limited future information against the optimal decisions with complete future information.

In this paper, a novel prediction-aware online algorithm is provided for energy generation scheduling in microgrids that considers a prediction window. We note that it is non-trivial to design a competitive prediction-aware online algorithm. A straightforward approach is to use receding horizon control (RHC), which determines the best possible decisions based on only the predicted future information, but does not consider any future events beyond the prediction window. Hence, RHC is not robust against the uncertainty beyond prediction. Therefore, a more robust algorithm is needed that can both harness the predicted information and accommodate the uncertainty beyond prediction. Such an algorithm should be sufficiently general to consider a parameterized prediction window with any window size. In this paper, we consider the energy generation scheduling problem for microgrids, where one needs to decide when to switch energy supply between a cheaper local generator with startup cost and the costlier on-demand external grid, considering intermittent renewable generation and fluctuating demands. There have been a number of recent studies about online energy generation scheduling. In the previous study [33], a prediction-oblivious algorithm called CHASE has been proposed to solve this problem. It is shown that CHASE achieves a competitive ratio of 3, which is the best among all deterministic online

algorithms. More generally, there is an abstract framework called Metrical Task System (MTS) problem [2], which considers general online decision-making processes for state changes with uncertain future switching costs among the states. We note that the online energy generation scheduling problem belongs to a class of scalar MTS problems, where the states are the number of generators being on (or off). However, there is no prediction-aware online algorithm for MTS in the literature so far, to the best of our knowledge. In this paper, we present a novel prediction-aware online algorithm with the best competitive ratio to date; see Sec. 2 for the discussion. Our algorithm not only solves the online energy generation scheduling problem, but also paves the way of tackling more general MTS problems with limited predicted information. MTS is capable of modeling many problems arising in a wide range of applications, including embedded systems and data centers [23], transportation systems [11], and online learning [1]. As another novelty considered in this work, the previous study [33] focuses on a homogeneous setting of local generators. In practice, however, microgrids may employ different types of generators with heterogeneous operating constraints. In this paper, we consider a more general setting where local generators can be heterogeneous with different capacities. We summarize our main contributions as follows:

(1) We propose CHASEpp as a novel prediction-aware online algorithm that can further improve the competitive ratio of the state-of-the-art CHASElk. This algorithm achieves competitive ratio of $3 - \left(2\alpha + 2(1-\alpha)/(1 + O(\frac{1}{w}))\right) \leq 3 - 2/(1 + O(\frac{1}{w}))$, where $\alpha \in [0, 1]$ is the system parameter that captures price discrepancy between using local generation and external sources to supply energy. Our algorithm achieves the best competitive ratio to date with up to 20% improvement than the state-of-the-art CHASElk. This competitive ratio also decreases twice faster with respect to w than CHASElk. We explore a new design space in our algorithm called cumulative differential cost in the prediction window, to better utilize the prediction information in making more competitive decisions. Our approach proactively monitors the possible online to offline cost ratio in the prediction window and makes intelligent online decisions.

(2) We also characterize a non-trivial lower bound of the competitive ratio. To obtain the lower bound, we create an adversary that progressively generates a worst-case input for any algorithm. We assume at anytime the accurate prediction of a future window is available to the algorithm. This means the adversary needs to build a window of input $\left(\left[\sigma(\tau)\right]_t^{t+w}\right)$ without knowing the algorithm's behavior in the upcoming window, which makes it difficult to establish the lower bound. In Sec. 9, we show that the competitive ratio of CHASEpp is close to the lower bound. For example, they only differ by 9% (i.e., 1.94 vs. 1.75) when we have a few hours of predictions.

(3) We use both theoretical analysis and trace-driven experiments to evaluate the performance of our algorithm by comparing it with the state-of-the-art algorithms. We also elaborate on how the input structure and system parameters can affect its performance. We note that our approach, with both perfect and noisy prediction information, can be extended to the online algorithm design for a general class of MTS problems with a similar structure.

Reference	Structure Exploitation	Competitive Ratio	Lower Bound	Heterogeneous Generators
Lin *et al.* [32]	✗	arbitrarily large (in a more general setting)	✗	✗
Hajiesmaili *et al.* [19]	✗	heuristic	✗	✗
Lu *et al.* [33]	✗	sub-optimal (partial use of the information)	✗	✗
This work	✓	reduces twice faster than [33] with w	✓	✓

Table 1: Summary and comparison of existing works.

2 RELATED WORK

Generation scheduling problems have attracted considerable attention. In large-scale power systems, with high aggregation effect on demand and small percentage of the erratic renewable generation, predicting the demand in the entire time horizon with a good level of accuracy is possible. Therefore the energy generation scheduling is basically an offline problem. Two main forms of this problem are economic dispatching [9, 14, 42] and unit commitment [16, 24, 40]. In the literature, researchers tackled this problem in different ways including dynamic programming [46], stochastic programming [47], and mixed-integer programming [5]. In recent years, with the increasing integration of the highly fluctuating renewable sources and deployment of small-scale microgrids, the uncertainty and intermittency have increased substantially on both supply and demand sides, and local supply-demand matching has become an essential part of the microgrid operation. 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 [25, 52]. The microgrid operator is the responsible party for local power balancing, which determines the optimal power generation and scheduling of all on-site resources [27]. To address the supply-demand matching problem in microgrids, researchers proposed different approaches which aim at scheduling either dispatchable generation on the supply side [36] or flexible load on the demand side (e.g., [6, 22]). Some other works combined these two with energy storage management [10, 17], in order to achieve power balance in microgrid.

In recent years, online optimization has emerged as a foundational topic in a variety of computer systems. It has seen considerable attention from applications in a wide range of research, including networking and distributed systems [38, 49, 50]. In [36], online convex optimization (OCO) framework [54] is used to design algorithms for the microgrid economic dispatch. OCO is a prominent paradigm being increasingly applied in different applications [3, 4, 7, 8, 30, 43]. There are some similarities between OCO with switching cost for dynamic scaling in datacenters [31] and the one of energy generation [33]. However, the inherent structures of both problems and solutions are significantly different. First, OCO considers a continuous feasible region, while in our setting with energy generators, the decision variable can only take discrete values. Second, their solution only applies to the homogeneous setting, while our solution can utilize multiple heterogeneous local generation units. Finally, in their recent work [32], the competitive ratio is sub-optimal compared to the existing solutions and our new algorithm. By increasing the switching cost, their competitive ratio increases linearly, while our algorithm's competitive ratio is always upper bounded by a constant that is independent of the switching cost. Other prediction-aware online algorithms like the one in [28]

also produce competitive ratios that grow unbounded as the switching cost increases. Some recent works [44, 45] tried to solve this issue by designing online algorithms with bounded competitive ratios. Still, their algorithm can only leverage prediction for large enough window sizes $w \geq r_{co}$, where r_{co} is a constant that grows unbounded as the switching cost increases. For example, in a real-world setting used in our numerical experiments (Sec. 9), for $w < 11$ hours, their competitive ratio is a constant that is independent of the window size. Meanwhile, the competitive ratio of our algorithm always keeps decreasing as the window size increases. In [53], a competitive algorithm design approach is used to solve the online economic dispatching problem with a peak-based charging model, which does not take the startup cost into account. The study in [33] incorporates the startup cost and turns the problem into a joint unit commitment and economic dispatch problem. In [19], a randomized online algorithm is proposed to solve this problem. In this paper, we aim to solve this problem with accurate prediction of the near future demand. In [33], 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 novel competitive online algorithm that will further improve both theoretical and practical performance over the previous algorithm. There are several aspects both in algorithm design and theoretical analysis that make our work to be different from other online solutions. We compare the most important aspects of these works and our work in Table 1.

3 ENERGY GENERATION SCHEDULING PROBLEM

The objective of energy generation scheduling in microgrids 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). We consider a system that operates in a time-slotted fashion, where $\mathcal{T}$ is the set of time slots, and the total length of the time horizon is T time slots ($T \triangleq |\mathcal{T}|$). The key notations are presented in Table 2.

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.

External grid and heating: 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 $p(t) \in [P_{\min}, P_{\max}]$. To keep the generality of the problem, we do not assume any specific stochastic model for the input profile $\sigma(t) \triangleq (a(t), h(t), p(t))$. Finally, to cover the heating demand, we can use external natural gas, costing c_g per unit of demand.

Local generation: We consider a heterogeneous setting where the power output capacity for power generation $n \in [1, N]$ is L_n, and these capacities can be different from each other. This generalizes the homogeneous setting considered in [33], where local generators have identical capacities. By adopting the widely-used

generator model [24], 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 [33], a general problem that includes these additional constraints is considered, and the approach to solve them is also proposed. In this paper, 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 using the same approach as in [33]. 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 $c_o \geq \eta \cdot c_g$, which means it is cheaper to obtain heat by using natural gas than purely by generators. To keep the problem interesting, we assume that $c_o + \frac{c_m}{L} \leq p_{\max} + \eta \cdot c_g$. 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. In this paper, we do not consider using energy storage in the generation scheduling problem. The reason is that for the typical size of microgrids, e.g., a college campus, existing energy storage systems are rather expensive and not widely available [35].

3.2 Problem Formulation

Let $v(t)$ and $s(t)$ be the amount of electricity and heat obtained from the external grid and the external natural gas at time t, respectively. Let $y_n(t)$ be the generator binary on/off status (1 as on and 0 as off), and $u_n(t)$ be the power output level of the n-th generator. The microgrid aggregated operational cost over the time horizon $\mathcal{T}$ is given by

$$\text{cost}(y, u, v, s) \triangleq \sum_{t \in \mathcal{T}} \Big(p(t)v(t) + c_g s(t) + \tag{1}$$

$$\sum_{n=1}^{N} \big[c_o u_n(t) + c_m y_n(t) + \beta [y_n(t) - y_n(t-1)]^+ \big] \Big),$$

that includes the grid electricity, external gas costs, and cost of the local generators, which is calculated by adding their operational cost and their switching cost over the entire time horizon $\mathcal{T}$. In this paper, we assume the initial status of all generators is off, i.e., $y_n(0) = 0$. Given the cost function and decision variables we formulate the **Microgrid Cost Minimization Problem** (MCMP) as follows:

$$\min_{y,u,v,s} \quad \text{cost}(y, u, v, s) \tag{2a}$$

$$\text{s.t.} \quad 0 \leq u_n(t) \leq L_n y_n(t), \tag{2b}$$

$$\sum_{n=1}^{N} u_n(t) + v(t) = a(t), \tag{2c}$$

$$\eta \cdot \sum_{n=1}^{N} u_n(t) + s(t) \geq h(t), \tag{2d}$$

$$\text{vars.} \quad y_n(t) \in \{0, 1\}, u_n(t), v(t), s(t) \in \mathbb{R}_0^+, n \in [1, N], t \in \mathcal{T},$$

Notation		Definition
Generator	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	$\mathcal{T}$	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. Var	$y(t)$	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 2: Key Notations. Brackets indicate the unit. We denote a vector by a single symbol, *e.g.*, $a \triangleq \left[a(t)\right]_{t=1}^{T}$.

where the constraint (2b) captures the capacity limit of the generators and the constraints (2c)-(2d) assure the electricity and heat demands are covered using the grid, natural gas, and the generators. It should be noted that the constraint (2c) is in the form of equality, which means that the electricity power-balance constraint is strictly satisfied. To ensure this, we run the local CHP generators, which might produce a heating supply that is more than the demand. There are various mechanisms to manage the excessive generated heat, including thermal storage systems coupled with CHP units, which allow storing energy and reusing it later by lowering the temperature of a substance, such as water [13].

We note that the AC Optimal Power Flow (OPF) constraints in the microgrid are not considered in MCMP, which is a joint unit commitment and economic dispatch problem. For certain microgrids, the ACOPF and economic dispatch should be coupled. However, if the microgrid is relatively large, similar to the conventional electric grids, to reduce the computational complexity, first unit commitment and economic dispatch are solved in hour-ahead time-scales, and then optimal power flow is solved minutes ahead of real-time [29]. On the other hand, for relatively small-scale microgrids, because of the short distances, negligible losses, and large line capacities, the constraints of the ACOPF problem will not be activated, and usually, the generation cost has the dominant impact on microgrid planning. Therefore, although in some microgrids with fast-responding generators, the economic dispatch and the ACOPF are solved together, for relatively small or relatively large

microgrids, this is not the case. In a general setting, the minimum turning on/off periods, and the ramping up/down rates can also be formulated as additional constraints. In [33] the authors propose an approach, which obtains the solution to the general problem using the solution to the "fast-responding" generators setting. A simple heuristic is to first compute solutions using the online and offline algorithms without the constraints and then modify the solutions to respect the switching constraints. In this paper, we also focus on the "fast-responding" generators, but our offline and online algorithms can be easily updated to incorporate the switching constraints of the general case using the same approach. Note that this minimization problem is challenging to solve for several reasons. First, even in offline setting, this problem is a mixed-integer linear problem, which is generally difficult to solve. Second, the startup cost $\beta[y_n(t) - y_n(t-1)]^+$ term in the objective function makes decisions coupled across the time, hence the problem cannot be decomposed. Finally, the input profile $\sigma(t) \triangleq (a(t), h(t), p(t))$ arrives online and we may not know the complete future input. In this paper, we first consider a microgrid with a single generator and solve the MCMP. Later in Sec. 7, we extend the solution to the multiple generator scenario. Therefore, we drop the subscript n, and the problem MCMP reduces to the problem MCMP_s for a single generator. We also utilize a useful observation to simplify the formulation: if the on/off status is given, the startup cost is determined, and MCMP_s reduces to a timewise decoupled linear program. According to [33] given a fixed on/off status $\left(y(t)\right)_{t=1}^{T}$, the solution that minimizes $\text{cost}(y, u, v, s)$ is

$$u(t) = \begin{cases} 0, & \text{if} \quad p(t) + \eta \cdot c_g \leq c_o, \\ \min\left\{\frac{h(t)}{\eta}, a(t), Ly(t)\right\}, & \text{if} \quad p(t) < c_o < p(t) + \eta \cdot c_g, \\ \min\left\{a(t), Ly(t)\right\}, & \text{if} \quad c_o \leq p(t), \end{cases}$$

$$v(t) = [a(t) - u(t)]^+, \text{ and } \quad s(t) = [h(t) - \eta \cdot u(t)]^+. \quad (3)$$

By using (3), the problem MCMP_s can be further simplified to the following problem with a single decision variable to turn on ($y(t) = 1$) or off ($y(t) = 0$) the generator.

$$\text{MCMP}_s: \quad \min_{y} \sum_{t \in \mathcal{T}} \left\{ \psi(\sigma(t), y(t)) + \beta[y(t) - y(t-1)]^+ \right\}$$
$$\text{vars. } y(t) \in \{0, 1\}, t \in \mathcal{T},$$

where $\psi(\sigma(t), y(t)) \triangleq p(t)v(t) + c_g s(t) + c_o u(t) + c_m y(t)$, and $(u(t), v(t), s(t))$ are defined based on the result in (3).

4 PRELIMINARY ON OFFLINE AND ONLINE SOLUTIONS

We first review state-of-the-art online solutions and the optimal offline solution for MCMP_s, providing necessary understandings towards designing a new algorithm later.

4.1 Optimal Offline Algorithm Design

In the offline setting the input $\left[\sigma(t)\right]_{t=1}^{T}$ is given at the beginning. We define

$$\delta(t) \triangleq \psi(\sigma(t), 0) - \psi(\sigma(t), 1), \quad (4)$$

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, to avoid turning on/off

the generator too frequently, the cumulative cost difference $\Delta(t)$ is defined as

$$\Delta(t) \triangleq \min\left\{0, \max\{-\beta, \Delta(t-1) + \delta(t)\}\right\}, \qquad (5)$$

where the initial value is $\Delta(0) = -\beta$. Now that $\Delta(t)$ is defined, we divide the time horizon $\mathcal{T}$ into several disjoint sets called critical segments. As shown in Fig. 1, each critical point T_i^c is defined using a pair $(T_i^c, \tilde{T}_i^c)$ corresponds to an interval where $\Delta(t)$ goes from $-\beta$ to 0 or from 0 to $-\beta$, without touching the boundaries. Based on the boundary values of these critical segments, we classify them into four categories as follows:

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

In [33], the optimal offline solution of MCMP$_s$ is given by

$$y^\star(t) \triangleq \begin{cases} 1, & \text{if } t \text{ is in a type-1 segment,} \\ 0, & \text{otherwise.} \end{cases} \qquad (6)$$

After getting the generator on/off status $y^\star(t)$, we apply (3) to obtain the optimal $u(t)$, $v(t)$, and $s(t)$.

4.2 Online Algorithm Without Prediction

To evaluate the performance of the online algorithm, the competitive ratio is defined as follows:

Definition 1. Let $\mathcal{A}$ be an online algorithm for MCMP$_s$. Define

$$\mathrm{CR}(\mathcal{A}) \triangleq \max_\sigma \frac{\mathrm{Cost}(y_{\mathcal{A}})}{\mathrm{Cost}(y_{\mathrm{OFA}})}. \qquad (7)$$

It is the worst-case ratio of the online cost over the offline cost. We proceed by explaining the online algorithm CHASE [33] that is later used in designing our new prediction-aware online algorithm. Recall that in the offline setting, we can detect the beginning of each critical segment right after the process enters them and set $y(t)$ accordingly. However, in the online setting, with no future information, it is impossible to do so. But, as shown in Fig. 1, 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 its competitive ratio satisfies

$$\mathrm{CR}(\mathrm{CHASE}) \leq 3 - 2\alpha, \qquad (8)$$

where

$$\alpha \triangleq \frac{c_{\mathrm{o}} + c_{\mathrm{m}}/L}{p_{\max} + \eta c_{\mathrm{g}}} \in (0, 1] \qquad (9)$$

and no other deterministic online algorithm can achieve a better competitive ratio. The adversarial input $[\sigma(t) \triangleq (a(t), h(t), p(t))]_1^T$ that results in this worst-case competitive ratio for CHASE is the input that always tries to make the online algorithm incur the maximum cost. Therefore, when the generator is on, the adversary gives zero demand $a(t) = 0$, and when the generator is off, the adversary gives maximum demand $a(t) = L$ as the input, as follows:

$$a(t) = \begin{cases} L, & \text{if } y(t-1) = 0, \\ 0, & \text{if } y(t-1) = 1, \end{cases} \quad h(t) = \eta a(t), \text{ and } p(t) = p_{\max}. \quad (10)$$
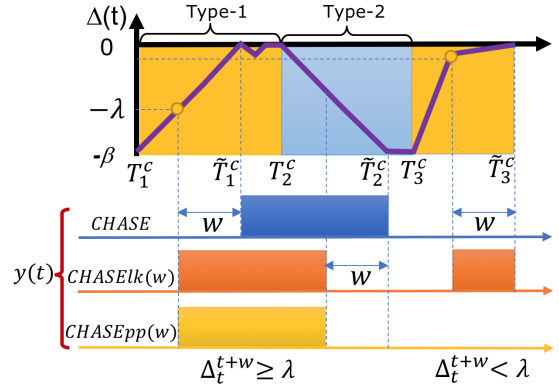

Figure 1: An example of $\Delta(t)$ and the online algorithms CHASE, CHASElk, and CHASEpp. The prediction-aware online algorithms detect the segment type w time slots before the prediction-oblivious CHASE.

Later in Sec. 6.1, when analyzing the competitive ratio of our new prediction-aware online algorithm, we also present its corresponding worst-case input.

4.3 Online Algorithm With Prediction

CHASE can be extended straightforwardly to the setting with prediction, where at each time slot t the precise prediction of the input for a window of w time slots $[\sigma(\tau)]_t^{t+w}$, is available. As one can see from the Fig. 1, the algorithm CHASElk(w) [33] behaves exactly the same as CHASE, except that it can detect the critical segment type and turn on/off the generator w time slots ahead of CHASE. Hence, CHASElk(w) achieves a better competitive ratio that satisfies

$$\mathrm{CR}(\mathrm{CHASElk}(w)) \leq 3 - 2f(\alpha, w) \leq 3 - 2\alpha, \qquad (11)$$

where

$$f(\alpha, w) = \alpha + \frac{(1-\alpha)}{1 + \beta \frac{Lc_{\mathrm{o}} + c_{\mathrm{m}}/(1-\alpha)}{wc_{\mathrm{m}}(Lc_{\mathrm{o}} + c_{\mathrm{m}})}} \qquad (12)$$

As discussed in Sec. 2, CHASElk(w) achieves the best competitive ratio with prediction prior to our study.

Next, we tackle this problem from a different perspective and propose a new threshold-based online algorithm that is substantially different from the existing algorithms. This prediction-aware online algorithm attains the best competitive ratio to date by exploiting a new design space.

5 NOVEL PREDICTION-AWARE ONLINE ALGORITHM

5.1 Intuitions

Consider the first and second type-1 critical segments in Fig. 1. For both of these segments the previous algorithm CHASElk(w) detects the segment type at $t = \tilde{T}_1^c - w$ and $t = \tilde{T}_3^c - w$, respectively and turns on the generator. Using these two examples, we explain two intuitions that motivate us to design better prediction-aware online algorithms.

- The first intuition is that in the first type-1 segment the cumulative cost difference in the window $[\tilde{T}_1^c - w, \tilde{T}_1^c]$ is large $(\Delta(\tilde{T}_1^c) - \Delta(\tilde{T}_1^c - w) = \lambda)$. This means there is a substantial

demand in the look-ahead window, and it is cost-effective to turn on the generator because spending the startup cost is worthy when we can enjoy the significant benefit of using the generator. Meanwhile, in the second type-1 segment, this cumulative cost difference is almost zero, which means sporadic demand in the look-ahead window. Hence, it is better to keep the generator off and avoid spending the high startup cost.

- The second intuition is that in the second type-1 segment, when $\Delta(\tilde{T}_3^c)$ reaches zero $\Delta(\tilde{T}_3^c - w)$ is almost zero and it means we already suffered from not turning on the generator earlier. Hence, turning on the generator at the current time $\tilde{T}_3^c - w$, when there is not enough demand in the look-ahead window, is not beneficial. On the other hand, in the first type-1 segment at time $\tilde{T}_1^c - w$, we have $\Delta(\tilde{T}_1^c - w) = -\lambda$, which means that we are still at the beginning of the type-1 segment and turning on the generator at this moment is worthy.

Following these intuitions, in the online setting, we turn on the generator only if we detect entering a type-1 critical segment, and there is a substantial benefit in using the generator in the look-ahead window. Meanwhile, as soon as we detect the type-2 critical segment, we should turn off the generator. Otherwise, the online algorithm will spend c_m unit per time slot of idling cost, while the offline has already turned off the generator and does not cost a penny. This 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 bought the ski, and its cost is fixed. To capture the benefit of using the generator, we define an important parameter named cumulative differential cost in the prediction window.

Definition 2. For any $\tau \in [t, t+w]$, we define Δ_t^τ as the *cumulative differential cost* between using or not using the generator in the time interval $[t, \tau]$ as follows:

$$\Delta_t^\tau \triangleq \sum_{s=t}^\tau \delta(s). \tag{13}$$

This parameter utilizes all the predicted information, and it is critical for our novel online algorithm design.

5.2 Algorithm Description

We denote our prediction-aware online algorithm as CHASEpp(w), as presented in Algorithm 1. In line 3 of the algorithm, if we detect being in a type-2 critical segment, we turn off the generator. Meanwhile, if we detect being in a type-1 critical segment (lines 5 to 13), we check for detecting the next type-2 critical segment. If we can not detect it by the end of the window (lines 7), we turn on the generator if we have $\Delta_t^{t+w} \geq \lambda$, which means there is enough benefit in using the generator for the prediction window. Similarly, if we can detect the next type-2 critical segment, and $\Delta_t^{\tau_2} \geq 0$ (line 9), we turn on the generator since there is enough benefit that can compensate the startup cost. Otherwise, we just keep the generator status unchanged.

We note that the online algorithm design space explored in this paper is new, in which turning on the generator depends on satisfying two conditions at the same time. First, we need to make

---

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

1: find $\Delta(\tau)_{\tau=t}^{t+w}$
2: set $\tau_1 \leftarrow \min\{\tau = t, ..., t+w \mid \Delta(\tau) = 0 \text{ or } -\beta\}$
3: **if** $\Delta(\tau_1) = -\beta$ (type-2) **then**
4:     $y(t) \leftarrow 0$
5: **else if** $\Delta(\tau_1) = 0$ (type-1) **then**
6:     set $\tau_2 \leftarrow \min\{\tau = t, ..., t+w \mid \Delta(\tau) = -\beta\}$
7:     **if** $\Delta(\tau) > -\beta, \forall \tau \in [t, t+w]$, and $\Delta_t^{t+w} \geq \lambda$ **then**
8:         $y(t) \leftarrow 1$
9:     **else if** $\Delta_t^{\tau_2} \geq 0$ **then**
10:        $y(t) \leftarrow 1$
11:     **else**
12:        $y(t) \leftarrow y(t-1)$
13:     **end if**
14: **else**
15:     $y(t) \leftarrow y(t-1)$
16: **end if**
17: set $u(t)$, $v(t)$, and $s(t)$ according to (3)

---

sure the offline algorithm has turned on the generator. Second, we turn on the generator only if there is a significant benefit in using it for the future window. By comparing CHASEpp(w) with the existing algorithms, one can see that the state-of-the-art algorithm CHASElk(w) is a simple extension of CHASE, which tracks the offline solution in an online manner. But with checking the cumulative differential cost in the prediction window, CHASEpp(w) makes smarter decisions when tracking the offline optimal to ensure better competitiveness. This new and effective design space improves the state-of-the-art competitive ratio.

6 PERFORMANCE ANALYSIS

6.1 Competitive Ratio

Let us denote the algorithm with $\lambda \geq 0$ as its threshold to be CHASEpp(w, λ). We have:

Theorem 1. *The competitive ratio of Algorithm 1 with $0 \leq \lambda \leq \beta$ is*

$$\text{CR}(\textit{CHASEpp}(w, \lambda)) = \max\{R_{\text{on}}(\lambda), R_{\text{off}}(\lambda)\}, \tag{14}$$

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

$$R_{\text{on}}(\lambda) = 1 + \left(1 - \frac{Lc_o + c_m}{L(p_{\max} + \eta \cdot c_g)}\right) \cdot \tag{15}$$

$$\max_{q \in \{0, wc_m\}} \left\{\frac{2\beta - q}{\beta + \left(2wc_m - q + \frac{c_o}{p_{\max} + \eta \cdot c_g}\lambda\right)\left(1 - \frac{c_m}{L(p_{\max} + \eta \cdot c_g - c_o)}\right)}\right\},$$

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

$$R_{\text{off}}(\lambda) = \frac{wc_m + \lambda}{wc_m + \frac{c_o}{p_{\max} + \eta \cdot c_g}\lambda}. \tag{16}$$

All the results presented in this paper have rigorous proofs, but due to page limit, we only provide a sketch of the idea behind our proof and present the details in Appendix A.

SKETCH OF PROOF. One can show that this algorithm has two and only two possible worst-case inputs. We explain each of these
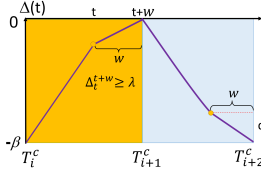
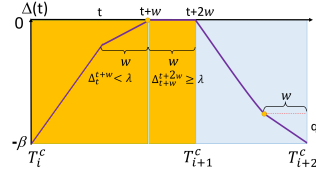Figure 2: First worst-case input for the type-1 and type-2 critical segments.

Figure 3: Second worst-case input appears by keeping the generator off for a window.

inputs, and the competitive ratio can be found by finding the maximum of the performance ratio among these two values.

First worst-case input: Consider the input in Fig. 2. The algorithm turns on the generator at $t = T_{i+1}^c - w$ and turns it off at $t = T_{i+2}^c - w$. We calculate performance ratio for this input as $R_{\text{on}}(\lambda)$ presented in (15), where $q = \Delta(T_{i+2}^c - w) - \Delta(T_{i+2}^c)$. This performance ratio is calculated by finding the maximum across the possible values of q.

Second worst-case input: Consider the example in Fig. 3. At time t we have $\Delta_t^{t+w} < \lambda$ and the algorithm keeps the generator off, but at time $t + w$ we have $\Delta_{t+w}^{t+2w} \geq \lambda$ and the algorithm turns on the generator. When calculating the performance ratio for this input, we observe that the ratio of the online cost increment over the offline cost increment is upper bounded by $R_{\text{off}}(\lambda)$ presented in (16). Hence, the competitive ratio is equal to the maximum of these two values.

□

6.2 The Optimal Threshold

We find the optimal threshold λ^* that minimizes the competitive ratio CR

$$\lambda^* = \arg\min_\lambda \max\{R_{\text{on}}(\lambda), R_{\text{off}}(\lambda)\}. \quad (17)$$

To understand how to find λ^* and its corresponding CR, we consider the example given in Fig. 4. In this example we observe that at $\lambda = 0$, we have $R_{\text{off}}(0) \leq R_{\text{on}}(0)$. Meanwhile, from Theorem 1 we know that $R_{\text{on}}(\lambda)$ is always a decreasing function and $R_{\text{off}}(\lambda)$ is always an increasing function. Hence, λ^* can be computed by finding the intersection of these two functions:

$$\lambda^* = \quad \max \quad \lambda \quad (18a)$$
$$\text{s.t.} \quad 0 \leq \lambda \leq \beta, \quad (18b)$$
$$0 \leq \lambda \leq L\left(p_{\max} + \eta \cdot c_g - c_o - \frac{c_m}{L}\right)w, \quad (18c)$$
$$R_{\text{on}}(\lambda) \geq R_{\text{off}}(\lambda), \quad (18d)$$

where (18b) ensures the threshold is not larger than the startup cost; otherwise, it is always optimal to turn on the generator. The constraint (18c) ensures that the threshold is within the maximum possible value, and (18d) ensures that we find the threshold that gives us the minimum competitive ratio. We can find the intersection of the two functions by using a simple binary search. As we can see in Fig. 5, by increasing the window size, the value of the threshold over the startup cost (λ^*/β) monotonically increases and approaches 1.

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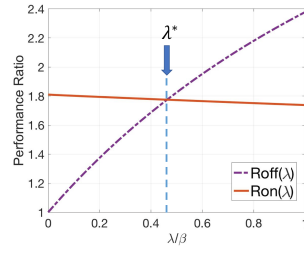

Figure 4: The value of λ^* is found at the intersection of the two functions $R_{\text{on}}(\lambda)$ and $R_{\text{off}}(\lambda)$.
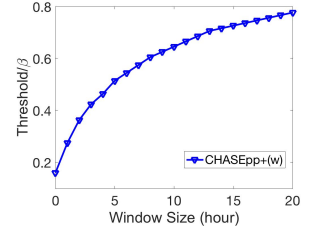
Figure 5: The value of the threshold over the startup cost β for different window sizes (w).

Theorem 2. *The competitive ratio of the algorithm CHASEpp(w) satisfies*

$$\text{CR} \leq 3 - 2g(\alpha, w), \quad (19)$$

where

$$g(\alpha, w) = \alpha + (1 - \alpha)\left(1 - \frac{1}{2} \cdot \max_{q \in \{0, wc_m\}}\left\{ \quad (20)\right.\right.$$
$$\left.\left.\frac{(2\beta - q)}{\beta + \left((2wc_m - q)(Lc_o + c_m) + \alpha Lc_o\lambda^*\right)/\left(Lc_o + c_m/(1 - \alpha)\right)}\right\}\right)$$

which is $\alpha + (1 - \alpha)/(1 + O(\frac{1}{w}))$ and monotonically increases from α to 1 as w increases.

PROOF. Refer to Appendix B. □

If there is no prediction ($w = 0$), we have $g(\alpha, 0) = \alpha$, and (19) reduces to the competitive ratio of CHASE. Meanwhile, if w is large, (20) turns to

$$g(\alpha, w) = \alpha+ \quad (21)$$
$$\frac{(1 - \alpha)}{1 + \beta(Lc_o + c_m/(1 - \alpha))/\underbrace{(2wc_m(Lc_o + c_m) + \alpha Lc_o\lambda^*)}_{(\dagger)}}.$$

One can see that by increasing the value of λ^* or w, value of the function $g(\alpha, w)$ keeps increasing thus, the competitive ratio keeps decreasing. To understand how the new algorithm improves the competitive ratio of CHASElk(w), we compare $g(\alpha, w)$ with $f(\alpha, w)$ presented in (11). One can see that in $g(\alpha, w)$, the term denoted as $(\dagger)$ is larger than the corresponding part in $f(\alpha, w)$ in (11) by $(wc_m(Lc_o + c_m) + \alpha Lc_o\lambda^*)$. This means that the competitive ratio decreases twice faster with respect to w than the previous algorithm. Therefore, our new algorithm always achieves better competitive ratio than the state-of-the-art algorithm CHASElk(w) as much as 20% improvement as shown in Fig. 9 to be discussed later.

Algorithms 2 CHASEpp$^+$(w)$[t, \sigma(\tau)_{\tau=t}^{t+w}, y(t-1)]$

1: **if** $\frac{1}{\alpha} < \text{CR}(\text{CHASEpp}(w))$ **then**
2: $\quad y(t) \leftarrow 0, \quad u(t) \leftarrow 0, \quad v(t) \leftarrow a(t), \quad s(t) \leftarrow h(t)$
3: $\quad$ return $(y(t), u(t), v(t), s(t))$
4: **else**
5: $\quad$ return CHASEpp(w)$[t, \sigma(\tau)_{\tau=t}^{t+w}, y(t-1)]$
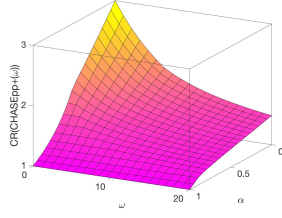6: **end if**

Figure 6: Competitive ratio of the algorithm $\text{CHASEpp}^+(w)$ as a function of α and w.

Note that from the definition, maximum value of $R_{\text{off}}(\lambda)$ is $1/\alpha$, which is the competitive ratio of the algorithm that only uses the external supply. Hence, when $\text{CR} = \max\{R_{\text{on}}(\lambda^*), R_{\text{off}}(\lambda^*)\} \geq 1/\alpha$, it means that instead of using our algorithm it is better to never turn on the generator. We summarize this result in Algorithm 2 denoted as $\text{CHASEpp}^+(w)$.

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

$$\text{CR}(\text{CHASEpp}^+(w)) = \min\{\text{CR}(\text{CHASEpp}(w)), \tfrac{1}{\alpha}\}. \qquad (22)$$

By the same logic $\text{CHASElk}^+(w)$ can be defined. In the rest of this paper, we evaluate the performance of these new algorithms. In Fig. 6, this competitive ratio is depicted as a function of α and w. When α is large, it means the economic advantage of using local generation over external sources is small. Hence, both online and offline algorithms tend to use local generation less often. This improves the competitiveness of online algorithms. Thus, by increasing α from 0 to 1, the competitive ratio decreases from 3 to 1 monotonically.

7 MULTIPLE GENERATOR SCENARIO

In this section, we solve the multi-generator **MCMP** problem presented in Sec. 3, where we have N units of heterogeneous generators. Without loss of generality, we assume that $L_1 \geq L_2 \ldots \geq L_N$. The key is to slice the demand into multiple layers, assign each sub-demand to a different generator, and solve an optimization problem for a single generator. For partitioning, we start from the bottom up, and we slice the demand such that the n-th layer has at most $a^{\text{ly}-\text{n}} = L_n$ units of electricity demand and $h^{\text{ly}-\text{n}} = \eta \cdot L_n$ units of heat demand. Once we used all the local generation capacities, for the remaining demands $(a^{\text{top}}, h^{\text{top}})$, we use the external sources. In this way, the bottom layers have the least frequent variations of demand, and we assign these layers to the generators with larger capacities. As a result, we use fewer generators, and these generators observe less variations, which helps to reduce the startup cost. The following theorem captures the performance of offline and online algorithms.

Theorem 3. *The offline algorithm that uses the layering approach produces an optimal offline solution for **MCMP**, and the online algorithm achieves the following competitive ratio:*

$$\text{CR} \leq 3 - 2g(\alpha_1, w), \qquad (23)$$

where $\alpha_1 = \frac{c_o + c_m / L_1}{p_{\max} + \eta c_g}$, and $g(\alpha, w)$ is defined in Theorem 2.
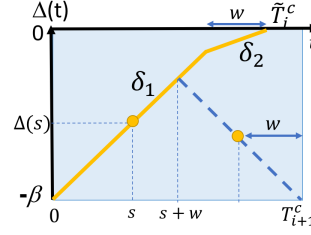
Proof. Refer to Appendix C. □
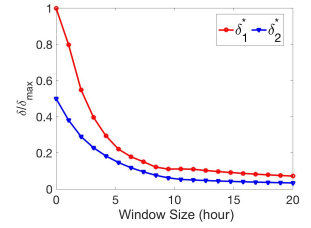

Figure 7: An example of the worst-case input with (δ_1, δ_2).

Figure 8: Value of (δ_1^*, δ_2^*), for different window sizes.

In the heterogeneous setting, each generator has its own unique prediction-aware online algorithm with a different optimal threshold that depends on their generation capacity L_n. The significance of this result is that it extends the applicability of our online algorithm beyond the homogeneous setting.

8 LOWER BOUND OF THE COMPETITIVE RATIO

Theorem 4. *Let $\epsilon > 0$ be the length of each time slot. As ϵ goes to zero and the discrete time setting approaches to the continues time setting, the competitive ratio for any prediction-aware deterministic online algorithm $\mathcal{A}$ for MCMP_s is lower bounded by*

$$\text{CR}(\mathcal{A}) \geq \text{cr}(w) = \frac{c_m + \delta_2^*}{c_m + \frac{Lc_o\alpha}{Lc_o + c_m}\delta_2^*}, \qquad (24)$$

where $\delta_2^* > 0$ is an optimal objective of an optimization problem with system parameters as input.

Proof. Refer to Appendix D. □

SKETCH OF PROOF. The key idea is that given any deterministic online algorithm $\mathcal{A}$, we progressively construct a particular worst-case input $\sigma(t) \triangleq (a(t), h(t), p(t))$, that for the performance ratio we have

$$\frac{\text{Cost}(y_{\mathcal{A}}; \sigma)}{\text{Cost}(y_{\text{OFA}}, \sigma)} \geq \text{cr}(w). \qquad (25)$$

In what follows, we explain this input, and it's corresponding lower bound in Lemma 1. Consider an example of the input in Fig. 7. Starting from the beginning $t = 0$, by giving a large value of differential cost ($\delta(t) = \delta_1$), the adversary tries to encourage the algorithm to turn on the generator. As soon as the algorithm turns on the generator at s, the adversary starts to hurt the algorithm as hard as possible by giving it $\delta(s + w) = -c_m$. Hence, the $\Delta(t)$ function keeps decreasing until the algorithm turns off the generator at time $t = T_{i+1}^c - w$. We denote the performance ratio for this input as $PR(s)$. One can see that at the beginning, the adversary gives a larger differential cost ($\delta(t) = \delta_1$) as the input. As time goes by and the offline cost changes, the adversary continues by giving a smaller differential cost δ_2 as input.

Lemma 1. *The lower bound of the competitive ratio is*

$$\text{CR}_l(\delta_1, \delta_2) = \min\{R_{\text{on}}(\delta_1, \delta_2), R_{\text{off}}(\delta_2)\}, \qquad (26)$$

where $R_{\text{on}}(\delta_1, \delta_2)$ is given by

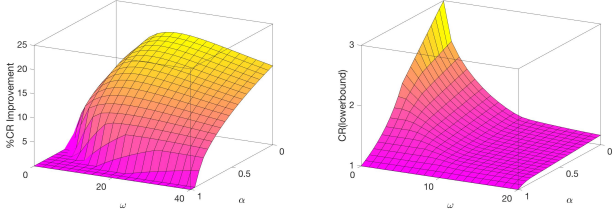$$R_{\text{on}}(\delta_1, \delta_2) = \min_{s \in [1, (\beta - w\delta_2)/\delta_1]} PR(s), \qquad (27)$$

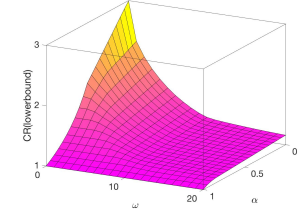Figure 9: Competitive ratio improvement as a function of α and w.

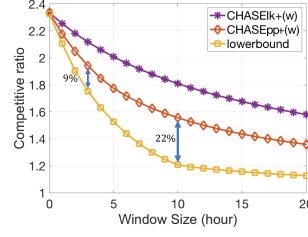Figure 10: Lower bound of the competitive ratio as a function of α and w.

Figure 11: Competitive ratio as a function of prediction window size (w).

Figure 12: Cost reduction as a function of prediction window size (w).

and $R_{\mathrm{off}}(\delta_2)$ is given by

$$R_{\mathrm{off}}(\delta_2) = \frac{c_{\mathrm{m}} + \delta_2}{c_{\mathrm{m}} + \frac{Lc_{\mathrm{o}}\alpha}{Lc_{\mathrm{o}}+c_{\mathrm{m}}}\delta_2}. \tag{28}$$

Proof. Refer to Appendix D.1. □

Similar to Theorem 1, we find (δ_1^*, δ_2^*) that maximizes the lower bound $\mathrm{CR}_{\mathrm{l}}(\delta_1, \delta_2)$. First, for each δ_2, we find a corresponding δ_1 such that $\delta_1 = \arg\max_{\delta} R_{\mathrm{on}}(\delta, \delta_2)$. This reduces $R_{\mathrm{on}}(\delta_1, \delta_2)$ to a single variable function of δ_2. Now we find the maximum of the minimum of two single variable functions. We know that for $\delta_2 = 0$, we have $R_{\mathrm{on}}(\delta_1, 0) \geq R_{\mathrm{off}}(0) = 1$, and $R_{\mathrm{off}}(\delta_2)$ is an increasing function. Hence, similar to (18) we keep increasing δ_2 until we find the intersection of the two functions. Therefore, we always have $R_{\mathrm{on}}(\delta_1^*, \delta_2^*) \geq R_{\mathrm{off}}(\delta_2^*)$ and $\mathrm{CR}_{\mathrm{l}}(\delta_1^*, \delta_2^*) = R_{\mathrm{off}}(\delta_2^*)$, which completes the proof. □

In Fig. 8, we plot (δ_1^*, δ_2^*) for different window sizes. As we can see, by increasing the window size, both δ_1^* and δ_2^* keep decreasing. As a result, the lower bound, which is a function of δ_2^*, keeps decreasing.

9 NUMERICAL EXPERIMENTS

We carry out numerical experiments using real-world traces to evaluate the performance of CHASEpp. 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 compared to this benchmark. We compare performance of the optimal offline algorithm OPT, CHASE, CHASElk$^+$, CHASEpp$^+$, and RHC, which is a popular algorithm widely used in the control literature [26], with both perfect and noisy prediction. The length of each time slot is 1 hour and the total cost incurred during one week (T = 168) is reported.

9.1 Experiment Setting

We obtain the electricity and heat demand traces from [12], which belongs 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 [37], which are collected from a wind farm outside San Francisco with an installed capacity of $12MW$. We obtain the electricity and natural gas price from PG&E [39], and we deploy generators with the same specifications as the one in [48], with heat recovery efficiency η set to be 1.8. 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
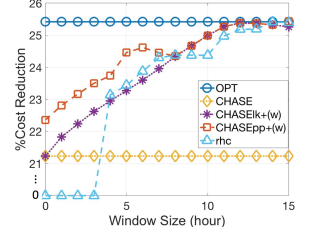
system with the unit heat generation cost of $c_{\mathrm{g}} = \$0.0179/KWh$, according to [15] and the startup cost β is set to be \$1400. The peak for the electricity demand is $30MW$, so we adopt 10 generators with maximum capacity $1MW \times 3$, $3MW \times 4$, and $5MW \times 3$ to fully satisfy the demand. All the experiments are modeled and implemented in Matlab [34] using Gurobi optimization tools [18].

9.2 Theoretical Ratio

In Fig. 9, we plot the competitive ratio improvement of our algorithm CHASEpp$^+(w)$ over CHASElk$^+(w)$ as a function of α and w. We can see that our algorithm improves the competitive ratio by up to 20%. As expected by decreasing the value of α, or increasing the window size, the competitive ratio improvement increases. But if we keep increasing the window size, both competitive ratios approach 1, and the competitive ratio gap starts to decrease. In a similar figure, the competitive ratio lower bound is depicted in Fig. 10. We can see that when w approaches 0, the lower bound approaches $3 - 2\alpha$, which is the lower bound of the prediction-oblivious CHASE. We also plot the competitive ratio and the lower bound for different values of w in Fig. 11. Our algorithm's competitive ratio is always better than the previous algorithm, and it is not far from the lower bound. With a 3-hour prediction, our algorithm's competitive ratio is away from the lower bound by 9%. In the worst case with $w = 10$ hours, our algorithm is away by at most 22%. One should note that in practice, 3 hours is a more typical prediction window size [41].

9.3 The Effect of Prediction Window

In this section, we change the window size from 0 to 15, and we show the results in Fig. 12. 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 CHASEpp$^+(w)$ performs better than the previous algorithm. It is important to note that depending on the input structure, there may be a large performance discrepancy between CHASElk$^+(w)$ and CHASEpp$^+(w)$. In the following section, we show how our new online algorithm can improve the previous algorithm's performance by exploiting the structure of the predicted information.

Effect of the cumulative differential cost: In this section, we build two inputs and depict the cost reduction of the two algorithms for these inputs in Fig. 13 and Fig. 14. We observe that in the first input shown in Fig. 13 with $a(t) = L$ when $\Delta(t + w)$ reaches zero the cumulative differential cost in the window is large, and hence both algorithms turn on the generator, and they have the same
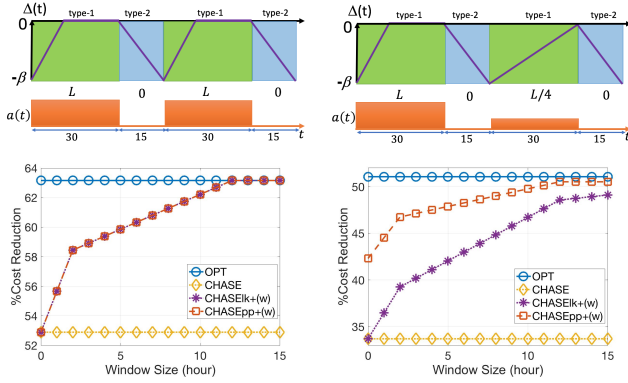
Figure 13: TInput with $a(t) \in \{0, L\}$**,** $h(t) = \eta a(t)$**, and** $p(t) = p_{max}$**, where both algorithms perform the same.**

Figure 14: Input with $a(t) \in \{0, L/4, L\}$**,** $h(t) = \eta a(t)$**, and** $p(t) = p_{max}$**, where the new algorithm performs better.**

Figure 15: Cost reduction for different sizes of the normal prediction error ($w = 1$**).**

Figure 16: Cost reduction for different sizes of the normal prediction error ($w = 3$**).**

Figure 17: Cost reduction for different sizes of the hyperbolic prediction error ($w = 1$**).**

Figure 18: Cost reduction for different sizes of the hyperbolic prediction error ($w = 3$**).**

performance. On the other hand, in the second input shown in Fig. 14 with $a(t) = L/4$ when $\Delta(t + w)$ reaches zero the cumulative differential cost is small, and the new algorithm does not turn on the generator and will not spend the additional startup cost, which leads to its better performance. Therefore, for a general input, if we have a large value of demand $a(t) = L$ coming in every time slot, both algorithms perform very well. But if the demand is small $a(t) = L/4$, the cumulative differential cost is also small, and using our new algorithm significantly improves the performance.

9.4 The Effect of Prediction Error

While the day-ahead electricity demand prediction has an error range of $2 - 3\%$, the highly fluctuating nature of the wind power makes the next hour's prediction error to usually be around $20-40\%$ [51]. Therefore, it is important to see how the prediction error can affect our online algorithm's performance. We obtain real-world wind power forecasting error distributions from [20], where the mean and standard deviation of the errors are based on the typical forecasts in the U.S., and hyperbolic distribution is used to represent the error. In [21] it has been shown that hyperbolic distribution is superior to the normal distribution in capturing wind power forecasting error. Still, to compare these two, we generate wind power forecasting errors from both. We start with the real-world hyperbolic distribution and zero-mean Gaussian, and in each time slot, we add the errors to the actual values. We also increase the standard deviation by 0 to 100% of the total installed capacity and the total peak demand for the wind power error and the heat demand error, respectively. In Fig. 15 and 16, we report the average cost reduction of the algorithms over 100 runs for two different lookahead window sizes of 1 and 3 hours with Gaussian errors. In Fig. 17 and 18 the results of the simulation for the real-world prediction errors with hyperbolic distribution are shown. It is important to note that for a 3-hour prediction window size the errors are often in $20-40\%$ range [51]. Therefore, by increasing the standard deviation up to 100%, we are stress-testing the algorithm. When w is small, both algorithms are robust to the prediction error. When the window size increases, however, the previous algorithm becomes more sensitive, and its performance starts deteriorating. Because, for a large window size,
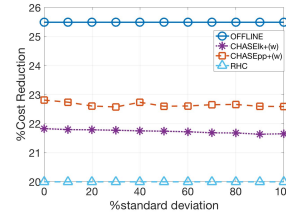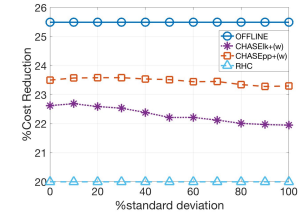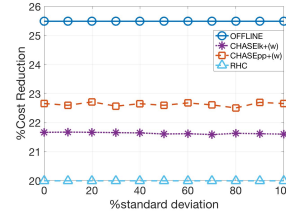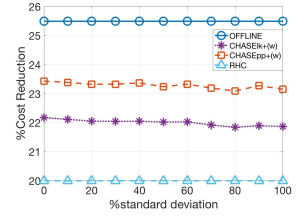
the prediction error of each time slot aggregates over the window, and if the window size becomes too large, the prediction can even worsen the algorithm's performance. On the other hand, the new online algorithm keeps its performance even for large prediction errors. The reason is that instead of only detecting the segment type, our algorithm checks the cumulative differential cost and only turns on the generator when it sees enough benefits in the look-ahead window.

10 CONCLUSION AND FUTURE WORK

We investigate how to leverage the prediction of the near future for online energy generation scheduling in microgrids. We tackle this problem from a new perspective and propose an effective online algorithm design that is fundamentally different from the existing algorithms. Our novel threshold-based online algorithm attains the best competitive ratio to date, which is upper-bounded by $3 - 2/(1 + O(\frac{1}{w}))$, where w is the prediction window size. We also characterize a non-trivial lower bound of the competitive ratio and show that the competitive ratio of our algorithm is only 9% away from the lower bound, when a few hours of prediction is available. Our theoretical and empirical evaluations demonstrate that our online algorithm outperforms the state-of-the-art ones. An interesting future direction is to exploit our new design space in developing competitive algorithms for general MTS problems with limited prediction. We also plan to incorporate energy storage system into the problem setting and algorithm design.

11 ACKNOWLEDGEMENT

REFERENCES

[1] Avrim Blum and Carl Burch. 2000. On-line learning and the metrical task system problem. *Machine Learning* 39, 1 (2000), 35–58.

[2] Allan Borodin and Ran El-Yaniv. 2005. *Online computation and competitive analysis.* Cambridge University Press.

[3] Xuanyu Cao and KJ Ray Liu. 2018. On the time-varying constraints and bandit feedback of online convex optimization. In *2018 IEEE International Conference on Communications (ICC).* IEEE, 1–6.

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

[5] Miguel Carrión and José M Arroyo. 2006. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on power systems* 21, 3 (2006), 1371–1378.

[6] Tsung-Hui Chang, Mahnoosh Alizadeh, and Anna Scaglione. 2013. Real-time power balancing via decentralized coordinated home energy scheduling. *IEEE Transactions on Smart Grid* 4, 3 (2013), 1490–1504.

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

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

[9] Po-Hung Chen and Hong-Chan Chang. 1995. Large-scale economic dispatch by genetic algorithm. *IEEE transactions on power systems* 10, 4 (1995), 1919–1926.

[10] Shengbo Chen, Ness B Shroff, and Prasun Sinha. 2013. Heterogeneous delay tolerant task scheduling and energy management in the smart grid with renewable energy. *IEEE Journal on Selected Areas in Communications* 31, 7 (2013), 1258–1267.

[11] Christian Coester and Elias Koutsoupias. 2018. The Online k-Taxi Problem. arXiv:1807.06645 [cs.DS]

[12] California Energy Commission. 2021. California Commercial End-Use Survey. https://www.energy.ca.gov/data-reports/surveys/california-commercial-end-use-survey [Online].

[13] Laura Fiorini and Marco Aiello. 2019. Energy management for user's thermal and power needs: A survey. *Energy Reports* 5 (11 2019), 1048–1076.

[14] Zwe-Lee Gaing. 2003. Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE transactions on power systems* 18, 3 (2003), 1187–1195.

[15] Green Energy UK. 2021. Gas and Electricity Price Quote. https://www.greenenergyuk.com/quote [Online].

[16] Xiaohong Guan, Qiaozhu Zhai, and Alex Papalexopoulos. 2003. Optimization based methods for unit commitment: Lagrangian relaxation versus general mixed integer programming. In *2003 IEEE Power Engineering Society General Meeting (IEEE Cat. No. 03CH37491)*, Vol. 2. IEEE, 1095–1100.

[17] Yuanxiong Guo, Miao Pan, Yuguang Fang, and Pramod P Khargonekar. 2013. Decentralized coordination of energy utilization for residential households in the smart grid. *IEEE transactions on smart grid* 4, 3 (2013), 1341–1350.

[18] Gurobi Optimization, LLC. 2022. Gurobi Optimizer Reference Manual.

[19] Mohammad H Hajiesmaili, Chi-Kin Chau, Minghua Chen, and Longbu Huang. 2016. Online microgrid energy generation scheduling revisited: The benefits of randomization and interval prediction. In *Proceedings of the Seventh International Conference on Future Energy Systems.* 1–11.

[20] Bri-Mathias Hodge, Debra Lew, Michael Milligan, Hannele Holttinen, S Sillanpaa, Emilio Gómez-Lázaro, Richard Scharff, L Soder, Xiaoli Guo Larsén, Gregor Giebel, et al. 2012. *Wind power forecasting error distributions: An international comparison.* Technical Report. National Renewable Energy Lab, Golden, CO (United States).

[21] Bri-Mathias S Hodge, Erik G Ela, and Michael Milligan. 2012. Characterizing and modeling wind power forecast errors from operational systems for use in wind integration planning studies. *Wind Engineering* 36, 5 (2012), 509–524.

[22] Yingsong Huang, Shiwen Mao, and R Mark Nelms. 2014. Adaptive electricity scheduling in microgrids. *IEEE Transactions on Smart Grid* 5, 1 (2014), 270–281.

[23] Sandy Irani, Sandeep K. Shukla, and Rajesh K. Gupta. 2003. Online strategies for dynamic power management in systems with multiple power-saving states. *ACM Trans. Embed. Comput. Syst.* 2, 3 (2003), 325–346.

[24] Spyros A Kazarlis, AG Bakirtzis, and Vassilios Petridis. 1996. A genetic algorithm solution to the unit commitment problem. *IEEE transactions on power systems* 11, 1 (1996), 83–92.

[25] Benjamin Kroposki. 2017. Integrating high levels of variable renewable energy into electric power systems. *Journal of Modern Power Systems and Clean Energy* 5, 6 (2017), 831–837.

[26] Wook Hyun Kwon and Allan E Pearson. 1977. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Trans. Automat. Control* 22, 5 (1977), 838–842.

[27] Jie Li, Yikui Liu, and Lei Wu. 2018. Optimal Operation for Community-Based Multi-Party Microgrid in Grid-Connected and Islanded Modes. *IEEE Transactions on Smart Grid* 9, 2 (2018), 756–765. https://doi.org/10.1109/TSG.2016.2564645

[28] Yingying Li, Guannan Qu, and Na Li. 2018. Online Optimization with Predictions and Switching Costs: Fast Algorithms and the Fundamental Limit. arXiv:1801.07780

[29] Zhongwen Li, Zhiping Cheng, Jing Liang, Jikai Si, Lianghui Dong, and Shuhui Li. 2020. Distributed Event-Triggered Secondary Control for Economic Dispatch and Frequency Restoration Control of Droop-Controlled AC Microgrids. *IEEE Transactions on Sustainable Energy* 11, 3 (2020), 1938–1950.

[30] Minghong Lin, Zhenhua Liu, Adam Wierman, and Lachlan L. H. Andrew. 2012. Online algorithms for geographical load balancing. In *2012 International Green Computing Conference (IGCC).* 1–10. https://doi.org/10.1109/IGCC.2012.6322266

[31] Minghong Lin, Adam Wierman, Lachlan L. H. Andrew, and Eno Thereska. 2011. Dynamic right-sizing for power-proportional data centers. In *2011 Proceedings IEEE INFOCOM.* 1098–1106. https://doi.org/10.1109/INFCOM.2011.5934885

[32] Yiheng Lin, Gautam Goel, and Adam Wierman. 2020. Online Optimization with Predictions and Non-Convex Losses. *Proc. ACM Meas. Anal. Comput. Syst.* 4, 1, Article 18 (May 2020), 32 pages.

[33] Lian Lu, Jinlong Tu, Chi-Kin Chau, Minghua Chen, and Xiaojun Lin. 2013. Online energy generation scheduling for microgrids with intermittent energy sources and co-generation. *ACM SIGMETRICS Performance Evaluation Review* 41, 1 (2013), 53–66.

[34] MATLAB. 2021. *version 9.10.0 (R2021a).* The MathWorks Inc.

[35] Ali Menati and Le Xie. 2021. A Preliminary Study on the Role of Energy Storage and Load Rationing in Mitigating the Impact of the 2021 Texas Power Outage. In *2021 North American Power Symposium (NAPS).* IEEE, 1–5.

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

[37] National Renewable Energy Laboratory. 2021. National Wind Technology Center's Information Portal. http://wind.nrel.gov [Online].

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

[39] Pacific Gas & Electric Company. 2021. Rate Information. http://www.pge.com/nots/rates/tariffs/rateinfo.shtml [Online].

[40] Narayana Prasad Padhy. 2004. Unit commitment-a bibliographical survey. *IEEE Transactions on power systems* 19, 2 (2004), 1196–1205.

[41] Madasthu Santhosh, Chintham Venkaiah, and DM Vinod Kumar. 2020. Current advances and approaches in wind speed and wind power forecasting for improved renewable energy integration: A review. *Engineering Reports* 2, 6 (2020), e12178.

[42] A Immanuel Selvakumar and K Thanushkodi. 2007. A new particle swarm optimization solution to nonconvex economic dispatch problems. *IEEE transactions on power systems* 22, 1 (2007), 42–51.

[43] Ming Shi, Xiaojun Lin, Sonia Fahmy, and Dong-Hoon Shin. 2018. Competitive Online Convex Optimization with Switching Costs and Ramp Constraints. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications.* 1835–1843.

[44] Ming Shi, Xiaojun Lin, and Lei Jiao. 2019. On the Value of Look-Ahead in Competitive Online Convex Optimization.

[45] Ming Shi, Xiaojun Lin, and Lei Jiao. 2021. Combining regularization with look-ahead for competitive online convex optimization. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications.* IEEE, 1–10.

[46] Walter L Snyder, H David Powell, and John C Rayburn. 1987. Dynamic programming approach to unit commitment. *IEEE Transactions on Power Systems* 2, 2 (1987), 339–348.

[47] Samer Takriti, John R Birge, and Erik Long. 1996. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems* 11, 3 (1996), 1497–1508.

[48] Tecogen. 2021. Cogeneration Product Data. https://www.tecogen.com/resources?media=product-data [Online].

[49] Jinlong Tu, Lian Lu, Minghua Chen, and Ramesh K Sitaraman. 2013. Dynamic provisioning in next-generation data centers with on-site power production. In *Proceedings of the fourth international conference on Future energy systems.* 137–148.

[50] Rahul Urgaonkar, Bhuvan Urgaonkar, Michael J Neely, and Anand Sivasubramaniam. 2011. Optimal power cost management using stored energy in data centers. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems.* 221–232.

[51] Jujie Wang and Yaning Li. 2018. Multi-step ahead wind speed prediction based on optimal feature extraction, long short term memory neural network and error correction strategy. *Applied Energy* 230 (2018), 429–443.

[52] Jingjie Yang, Bingqing Guo, and Bo Qu. 2018. Economic optimization on two time scales for a hybrid energy system based on virtual storage. *Journal of Modern Power Systems and Clean Energy* 6, 5 (2018), 1004–1014.

[53] Ying Zhang, Mohammad H Hajiesmaili, Sinan Cai, Minghua Chen, and Qi Zhu. 2016. Peak-aware online economic dispatching for microgrids. *IEEE transactions on smart grid* 9, 1 (2016), 323–335.

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

A PROOF OF THEOREM 1

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 $\theta \in [1, w]$. $k = 0$ 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 CHASEpp(w) by $\left(y_{\text{CHASE}(w)}(t)\right)_{t=1}^T$, and the outcome of the optimal offline algorithm by $\left(y_{OFA}(t)\right)_{t=1}^T$. Let us define $\mathcal{K}_j$ as the set of indices of type-j critical segments i.e.,

$$\mathcal{K}_j \triangleq \{\, i \mid [T_i^c + 1, T_{i+1}^c]\text{ is a type-}j\text{ critical segment in } [0, T]\}.$$

Denote the sub-cost for type-j by

$$\text{Cost}^{\text{ty-}j}(y) \triangleq \sum_{i \in \mathcal{K}_j} \sum_{t=T_i^c+1}^{T_{i+1}^c} \psi\left(\sigma(t), y(t)\right) + \beta \cdot [y(t) - y(t-1)]^+ \quad (29)$$

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

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

$$\text{Cost}^{\text{ty-0}}(y_{OFA}) = \text{Cost}^{\text{ty-0}}(y_{\text{CHASE}(w)}) \quad (30)$$

(Type-1): Based on the definition of critical segment, we recall that there is an auxiliary point $\tilde{T}_i^c$, such that either $\left(\Delta(T_i^c) = 0 \text{ and } \Delta(\tilde{T}_i^c) = -\beta\right)$ or $\left(\Delta(T_i^c) = -\beta \text{ and } \Delta(\tilde{T}_i^c) = 0\right)$. 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

$$y_{\text{CHASE}(w)}(t) = \begin{cases} 0, & \text{for all } t \in [T_i^c + 1, \tilde{T}_i^c - \theta), \\ 1, & \text{for all } t \in [\tilde{T}_i^c - \theta, T_{i+1}^c]. \end{cases} \quad (31)$$

We consider a particular type-1 critical segment: $[T_i^c + 1, T_{i+1}^c]$. Note that by the definition of type-1, at the beginning of this segment for both online and offline algorithms the generator status is off $y_{OFA}(T_i^c) = y_{\text{CHASE}(w)}(T_i^c) = 0$. For the offline algorithm $y_{OFA}(t)$ switches from 0 to 1 at time $T_i^c + 1$, while for the online algorithm $y_{\text{CHASE}(w)}$ switches at time $\tilde{T}_i^c - \theta - 1$, both incurring startup cost β. Hence, in the interval $[T_i^c + 1, \tilde{T}_i^c - \theta - 1]$, the online and offline algorithms have a different status, while in the interval $[\tilde{T}_i^c - \theta, T_{i+1}^c]$ they have the same status. The cost difference between $y_{\text{CHASE}(w)}$ and y_{OFA} within $[T_i^c + 1, T_{i+1}^c]$ is

$$\sum_{t=T_i^c+1}^{\tilde{T}_i^c-1} \left(\psi\left(\sigma(t), 0\right) - \psi\left(\sigma(t), 1\right)\right) + \beta - \beta = \sum_{t=T_i^c+1}^{\tilde{T}_i^c-\theta-1} \delta(t)$$

$$= \Delta(\tilde{T}_i^c - \theta - 1) - \Delta(T_i^c) = -q_i^1 + \beta, \quad (32)$$

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

If we repeat this process for all type-1 critical segments, and we have $m_1 \triangleq |\mathcal{K}_1|$ type-1 critical segments, we obtain

$$\text{Cost}^{\text{ty-1}}(y_{\text{CHASE}(w)}) \leq \text{Cost}^{\text{ty-1}}(y_{OFA}) + m_1 \cdot \beta - \sum_{i \in \mathcal{K}_1} q_i^1. \quad (33)$$

(Type-2) and **(Type-3)**: Now, we repeat the same process for type-2 and type-3 (type-end) critical segments. Let the number of type-j critical segments be $m_j \triangleq |\mathcal{K}_j|$. We derive similarly for $j = 2$ or 3 as

$$\text{Cost}^{\text{ty-}j}(y_{\text{CHASE}(w)}) \leq \text{Cost}^{\text{ty-}j}(y_{OFA}) + m_j \cdot \beta - \sum_{i \in \mathcal{K}_j} q_i^j, \quad (34)$$

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

Note that $|q_i^j| \leq \beta$ 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

$$\frac{\text{Cost}(y_{\text{CHASE}(w)})}{\text{Cost}(y_{OFA})} = \frac{\sum_{j=0}^3 \text{Cost}^{\text{ty-}j}(y_{\text{CHASE}(w)})}{\sum_{j=0}^3 \text{Cost}^{\text{ty-}j}(y_{OFA})} \leq$$

$$1 + \frac{2m_1\beta + \sum_{i \in \mathcal{K}_1} q_i^1 - \sum_{i \in \mathcal{K}_2} q_i^2 - \sum_{i \in \mathcal{K}_3} q_i^3}{\sum_{j=0}^3 \text{Cost}^{\text{ty-}j}(y_{OFA})} \quad (35)$$

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

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

$$\leq 1 + \begin{cases} 0 & \text{if } m_1 = 0, \\ \dfrac{2m_1\beta - \sum_{i \in \mathcal{K}_1} q_i^1 - \sum_{i \in \mathcal{K}_2} q_i^2}{\sum_{j=0}^2 \text{Cost}^{\text{ty-}j}(y_{OFA})} & \text{otherwise} \end{cases} \quad (36)$$

By Lemma 2, and Lemma 3 and simplifications, we obtain:

$$\frac{\text{Cost}(y_{\text{CHASE}(w)})}{\text{Cost}(y_{OFA})} \leq 1 + \left(1 - \frac{Lc_o + c_m}{L(p_{\max} + \eta \cdot c_g)}\right) \cdot \max_{q \in \{0, wc_m\}}$$

$$\left\{\frac{(2\beta - q)}{\beta + \left(2wc_m - q + \frac{c_o}{p_{\max}+\eta\cdot c_g}\lambda\right)\left(1 - \frac{c_m}{L(p_{\max}+\eta\cdot c_g-c_o)}\right)}\right\} \quad (37)$$

We denote this value as $R_{\text{on}}(\lambda)$. 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 4, if we change to general k, the increment ratio is as follows:

$$\frac{\text{online cost increment}}{\text{offline cost increment}} \leq \frac{k\left(wc_m + \lambda\right)}{k\left(wc_m + \frac{c_o}{p_{\max}+\eta\cdot c_g}\lambda\right)} \quad (38)$$

We denote this ratio as $R_{\text{off}}(\lambda)$. If this ratio is larger than the previous one $R_{\text{off}}(\lambda) > R_{\text{on}}(\lambda)$, then by increasing k, the value of the performance ratio keeps increasing and when k goes to ∞, this competitive ratio goes to $R_{\text{off}}(\lambda)$. On the other hand, if $R_{\text{off}}(\lambda) \leq R_{\text{on}}(\lambda)$, by increasing the value of k, value of the competitive ratio will not increase and is still upper bounded by $R_{\text{on}}(\lambda)$. This shows that the competitive ratio is upper bonded by the maximum of $R_{\text{on}}(\lambda)$, and

$R_{\text{off}}(\lambda)$. In Lemma 5 we show that $R_{\text{off}}(\lambda)$ is always an increasing function while $R_{\text{on}}(\lambda)$ is always a decreasing function. This completes the proof. □

A.1 Proof of Lemma 2

Lemma 2. *For the (**type-1**), we have*

$$\text{Cost}^{\text{ty-1}}(y_{\text{OFA}}) \geq m_1\beta + \sum_{i \in \mathcal{K}_1} \Big(\frac{(q_i^1 + \beta)(Lc_{\text{o}} + c_{\text{m}})}{L(p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}) - c_{\text{m}}} +$$

$$\frac{p_{\max} + \eta \cdot c_{\text{g}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} \big(wc_{\text{m}} + \frac{c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}}} \lambda \big) \Big). \tag{39}$$

PROOF. Consider a particular type-1 segment $[T_i^c + 1, T_{i+1}^c]$. We denote its offline cost as Cost^{t1}. We have:

$$\text{Cost}^{\text{t1}} = \beta + \sum_{t=T_i^c+1}^{T_{i+1}^c} \psi(\sigma(t), 1) = \beta +$$

$$(T_{i+1}^c - T_i^c)c_{\text{m}} + \sum_{t=T_i^c+1}^{T_{i+1}^c} \big(\psi(\sigma(t), 1) - c_{\text{m}} \big). \tag{40}$$

By [33, Lemma. 4] and simplification we obtain

$$\text{Cost}^{\text{t1}} \geq \beta + (T_{i+1}^c - T_i^c)c_{\text{m}} + \frac{c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} \cdot$$

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

$$(T_{i+1}^c - T_i^c)c_{\text{m}} + \frac{c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} \sum_{t=T_i^c+1}^{T_{i+1}^c} \delta(t) \tag{41}$$

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

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

$$\sum_{t=T_i^c+1}^{T_{i+1}^c} \delta(t) = \sum_{t=T_i^c+1}^{\tilde{T}_i^c-\theta-1} \delta(t) + \sum_{\tilde{T}_i^c-\theta}^{T_{i+1}^c} \delta(t) = \Delta(\tilde{T}_i^c - \theta - 1) - \Delta(T_i^c)$$

$$+ \sum_{\tilde{T}_i^c-\theta}^{T_{i+1}^c} \delta(t) = \beta - q_i^1 + \sum_{\tilde{T}_i^c-\theta}^{T_{i+1}^c} \delta(t) \geq \beta - q_i^1 + \lambda$$

Step 2: To find the lower bound of the length of the interval $[T_i^c + 1, T_{i+1}^c]$, 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 $[T_i^c + 1, T_{i+1}^c]$ has two part as $[T_i^c + 1, \tilde{T}_i^c - w - 1]$ and $[\tilde{T}_i^c - w, T_{i+1}^c]$. We note that $(\tilde{T}_i^c - w - 1 - T_i^c)$ is lower bounded by the steepest descend when $p(t) = p_{\max}$, $a(t) = L$ and $h(t) = \eta L$,

$$\tilde{T}_i^c - w - 1 - T_i^c \geq \frac{\beta - q_i^1}{L(p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}) - c_{\text{m}}}, \tag{42}$$

and for the second part we have $T_{i+1}^c - \tilde{T}_i^c + w + 1 \geq w$, which means

$$T_{i+1}^c - T_i^c \geq \frac{\beta - q_i^1}{L(p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}) - c_{\text{m}}} + w \tag{43}$$

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 λ, hence to calculate the total $[T_i^c + 1, T_{i+1}^c]$ length we have

$$\sum_{t=T_i^c+1}^{T_{i+1}^c} \delta(t) \geq \beta - q_i^1 + \lambda \implies \sum_{t=T_i^c+1}^{T_{i+1}^c} \delta(t) - \sum_{\tilde{T}_i^c-w}^{\tilde{T}_i^c} \delta(t) + \quad (44)$$

$$\sum_{\tilde{T}_i^c-w}^{\tilde{T}_i^c} \delta(t) \geq \beta - q_i^1 + \lambda \implies \sum_{t=T_i^c+1}^{T_{i+1}^c} \delta(t) - \sum_{\tilde{T}_i^c-w}^{\tilde{T}_i^c} \delta(t) \geq \beta - q_i^1,$$

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

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

$$\implies T_{i+1}^c - T_i^c \geq \frac{\beta - q_i^1}{L(p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}) - c_{\text{m}}} + w \tag{45}$$

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

$$T_{i+1}^c - T_i^c \geq \frac{\beta - q_i^1}{L(p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}) - c_{\text{m}}} + w \tag{46}$$

length By Eqns. (42)-(46), we obtain

$$\text{Cost}^{\text{t1}} \geq \beta +$$

$$\frac{p_{\max} + \eta \cdot c_{\text{g}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} \big(\frac{\beta - q_i^1}{L(p_{\max} + \eta c_{\text{g}} - c_{\text{o}} - \frac{c_{\text{m}}}{L})} + w \big) c_{\text{m}} +$$

$$\frac{c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} (\beta - q_i^1 + \lambda) = \beta + \frac{(\beta - q_i^1)(Lc_{\text{o}} + c_{\text{m}})}{L(p_{\max} + \eta c_{\text{g}} - c_{\text{o}} - \frac{c_{\text{m}}}{L})}$$

$$+ \frac{p_{\max} + \eta \cdot c_{\text{g}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} \big(wc_{\text{m}} + \frac{c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}}} \lambda \big) \tag{47}$$

Since there are m_1 type-1 critical segments, according to Eqna. (47), we obtain

$$\text{Cost}^{\text{ty-1}}(y_{\text{OFA}}) \geq m_1\beta + \sum_{i \in \mathcal{K}_1} \Big(\frac{(\beta - q_i^1)(Lc_{\text{o}} + c_{\text{m}})}{L(p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}) - c_{\text{m}}}$$

$$+ \frac{p_{\max} + \eta \cdot c_{\text{g}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} \big(wc_{\text{m}} + \frac{c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}}} \lambda \big) \Big). \tag{48}$$

□

A.2 Proof of Lemma 3

Lemma 3. *For (**type-2**), we have*

$$\text{Cost}^{\text{ty-2}}(y_{\text{OFA}}) \geq \sum_{i \in \mathcal{K}_2} \Big(\frac{p_{\max} + \eta \cdot c_{\text{g}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} (w \cdot c_{\text{m}} - q_i^2) \Big) \tag{49}$$

PROOF. Consider a particular type-2 segment $[T_i^c + 1, T_{i+1}^c]$. We denote its offline cost as Cost^{t2}. We have:

$$\text{Cost}^{\text{t2}} = \sum_{t=T_i^c+1}^{T_{i+1}^c} \psi(\sigma(t), 0) \tag{50}$$

By [33, Lemma. 4] and simplification we obtain

$$\text{Cost}^{\text{t2}} \geq \frac{p_{\max} + \eta \cdot c_{\text{g}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} \left(\sum_{t=T_i^c+1}^{T_{i+1}^c} \delta(t) + (T_{i+1}^c - T_i^c) c_{\text{m}} \right)$$

$$= \frac{p_{\max} + \eta \cdot c_{\text{g}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} \left(-\beta + (T_{i+1}^c - T_i^c) c_{\text{m}} \right) \quad (51)$$

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

$$T_{i+1}^c - T_i^c \geq w + \frac{\beta - q_i^2}{c_{\text{m}}} \quad (52)$$

By Eqns. (51)-(52), we obtain

$$\text{Cost}^{\text{t2}} \geq \frac{p_{\max} + \eta \cdot c_{\text{g}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} (w \cdot c_{\text{m}} - q_i^2) \quad (53)$$

Since there are m_2 type-2 critical segments, according to Eqna. (53), we obtain

$$\text{Cost}^{\text{ty-2}}(y_{\text{OFA}}) \geq \sum_{i \in \mathcal{K}_2} \left(\frac{p_{\max} + \eta \cdot c_{\text{g}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} (w \cdot c_{\text{m}} - q_i^2) \right). \quad (54)$$

□

A.3 Proof of Lemma 4

Lemma 4. *Consider a window $[t, t+w]$, in the type-1 critical segment. If we have $\Delta_t^{t+w} \leq \lambda$, then the cost of the online algorithm over the cost of the optimal offline algorithm in this window is upper bounded by the following:*

$$\frac{\text{Cost}(y_{\text{CHASEpp}(w)})}{\text{Cost}(y_{\text{OFA}})} \leq \frac{wc_{\text{m}} + \lambda}{wc_{\text{m}} + \frac{c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}}} \lambda}. \quad (55)$$

PROOF. By [33, Lemma. 4] and simplification we know that in a type-1 critical segment for a window with $\Delta_t^{t+w} = \lambda$, for the offline cost we have

$$\text{Cost}(y_{\text{OFA}}) \geq \frac{p_{\max} + \eta \cdot c_{\text{g}}}{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}} \left(wc_{\text{m}} + \frac{c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}}} \lambda \right) \quad (56)$$

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 λ, which means

$$\text{Cost}(y_{\text{CHASEpp}(w)}) - \text{Cost}(y_{\text{OFA}}) = \lambda \quad (57)$$

Hence we have

$$\frac{\text{Cost}(y_{\text{CHASEpp}(w)})}{\text{Cost}(y_{\text{OFA}})} = 1 + \frac{\lambda}{\text{Cost}(y_{\text{OFA}})} \leq \frac{wc_{\text{m}} + \lambda}{wc_{\text{m}} + \frac{c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}}} \lambda} \quad (58)$$

This completes the proof. □

A.4 Proof of Lemma 5

Lemma 5. *$R_{\text{on}}(a)$ is always a decreasing function of a and $R_{\text{off}}(a)$ is always an increasing function of a.*

PROOF. $R_{\text{on}}(\lambda)$: To prove that $R_{\text{on}}(\lambda)$ is always a decreasing function first we take the derivative as a function of λ. To compute the derivative we only consider the maximization part of $R_{\text{on}}(\lambda)$.

The denominator of the function is always positive and the numerator is given by

$$-\left(\frac{(2\beta - q)c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}}} \lambda \right) \left(1 - \frac{c_{\text{m}}}{L(p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}})} \right). \quad (59)$$

Note that we have $q \in \{0, wc_{\text{m}}\}$. If $wc_{\text{m}} < 2\beta$ the derivative is always negative. If $wc_{\text{m}} \geq 2\beta$ 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 $wc_{\text{m}} \geq 2\beta$ 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

$$-\beta - \left(2(wc_{\text{m}} - \beta) + \frac{c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}}} \lambda \right) \left(1 - \frac{c_{\text{m}}}{L(p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}})} \right). \quad (60)$$

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

$R_{\text{off}}(\lambda)$: Now we show that $R_{\text{off}}(\lambda)$ 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

$$\frac{p_{\max} + \eta \cdot c_{\text{g}} - c_{\text{o}}}{p_{\max} + \eta \cdot c_{\text{g}}} wc_{\text{m}}, \quad (61)$$

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

B PROOF OF THEOREM 2

PROOF. From Theorem 1 we have

$$\text{CR}(\mathcal{A}(\lambda^*)) = \max\{R_{\text{on}}(\lambda^*), R_{\text{off}}(\lambda^*)\} \quad (62)$$

and from the definition of the optimal threshold λ^* in (18d) we have

$$\max\{R_{\text{on}}(\lambda^*), R_{\text{off}}(\lambda^*)\} = R_{\text{on}}(\lambda^*) \quad (63)$$

Therefore for the competitive ratio we have:

$$\text{CR}(\mathcal{A}(\lambda^*)) = R_{\text{on}}(\lambda^*). \quad (64)$$

By using the definition of α in (9) and simplification we obtain the result which completes the proof. □

C PROOF OF THEOREM 3

PROOF. This theorem includes two parts as follows:

- **Offline algorithm:** As shown in [33, Theorem. 5, and 6], when we have homogeneous generators, the offline algorithm that uses the layering approach produces an optimal offline solution for **MCMP**. For the case with multiple heterogeneous generators, since assigning the bottom layers to the generators with larger capacities minimizes the start-up cost, it can easily be shown that the layering approach also leads to an optimal offline solution. On the other hand, the operational cost does not depend on the capacity L_n, and it is the same for all the generators. Hence, it can be shown that the layering approach produces an offline optimal.

- **Online algorithm:** In this case, each generator is solving its own sub-problem with a given sub-demand. Hence, the competitive ratio of the algorithm is upper bounded by the largest competitive ratio among all generators.

$$\text{CR} \leq \max_{n \in [1,N]} 3 - 2g(\alpha_n, w), \tag{65}$$

where

$$\alpha_n = \frac{c_o + c_m/L_n}{p_{\max} + \eta c_g}. \tag{66}$$

Since the competitive ratio is an increasing function of L and $L_1 \geq L_2 ... \geq L_N$, we have:

$$\text{CR} \leq 3 - 2g(\alpha_1, w). \tag{67}$$

This completes the proof.

□

D PROOF OF THEOREM 4

PROOF. Finding the lower bound of the competitive ratio is equal to constructing an special input $\sigma(t) \triangleq (a(t), h(t), p(t))$ such that for any deterministic online algorithm $\mathcal{A}$, we have:

$$\frac{\text{Cost}(y_{\mathcal{A}}; \sigma)}{\text{Cost}(y_{\text{OFA}}, \sigma)} \geq \text{cr}(w). \tag{68}$$

In [33], it is shown that when we do not have any prediction $w = 0$, the following input gives us the lower bound.

$$\delta(t) = \begin{cases} \delta_{max}, & \text{if} \quad y(t-1) = 0, \\ \delta_{min}, & \text{if} \quad y(t-1) = 1. \end{cases} \tag{69}$$

As one can see in this input, as long as the algorithm keeps the generator off, the adversary keeps giving full demand (δ_{max}) as the input, and as soon as the algorithm turns off the generator, the adversary starts giving zero demand (δ_{min}) as the input. This simple input is designed in a way that it always tries to hurt the algorithm most. In creating the lower bound for the case with perfect prediction, we follow the same logic, but we need to design a different input.

If we keep giving full demand until the algorithm turns on the generator, at some point, we have already given a lot of full demand to the algorithm, and by turning on the generator, the algorithm can enjoy a window of full demand. In this way, we can not really hurt the algorithm. Therefore we need to carefully choose the demand in the future window in a way that it gives the algorithm some incentive to turn on the generator, but at the same time, it does not give it a lot of demand in the coming window to enjoy. By carefully adjusting this demand, we can find the lower bound.

At any time t, we need to construct the input of the time t, and the algorithm has already decided the generator status for the time $[1, t - w - 1]$. We need to know that for how many consecutive time slots the algorithm kept the generator off. To this end, we define a counter called $c(t)$. This counter will reset anytime we turn on the generator and keeps increasing when the algorithm keeps the generator off. We define it as follows:

$$c(t) = \begin{cases} 0, & \text{if} \quad y(t-w-1) = 1, \\ c(t-1) + 1, & \text{if} \quad y(t-w-1) = 0, \end{cases} \tag{70}$$

where the initial value of y we have $c(0) = 0$, and $y(t)_{t=-w}^{0} = 0$.

We construct the worst-case input as follows:

$$\delta(t) = \begin{cases} \delta_1, & \text{if} \quad c(t) \leq \frac{\beta - w\delta_2}{\delta_1}, \\ \delta_2, & \text{if} \quad c(t) > \frac{\beta - w\delta_2}{\delta_1}. \end{cases} \tag{71}$$

Now we need to calculate the proper values for δ_1 and δ_2. We use Lemma 1 toward this end.

D.1 Proof of Lemma 1

PROOF. Consider the input shown in Fig. 7. If the algorithm turns on the generator at some point $s \in [1, \frac{\beta - w\delta_2}{\delta_1}]$, we can calculate the performance ratio as the online cost over the offline cost as follows:

$$PR(s) = 1 + \frac{\beta - (q(s+w) - q(s)) + max(q(s+w) - wc_m, 0)}{\frac{p_{\max} + \eta \cdot c_g}{p_{\max} + \eta \cdot c_g - c_o}((s+w)c_m + q(s+w))}, \tag{72}$$

where $q(s) = \Delta(s) + \beta$. Since we are looking for the lower bound, we find the minimum across all possible values of s. We call this value $R_{on}(\delta_1, \delta_2)$ and define it as follows:

$$R_{on}(\delta_1, \delta_2) = \min_{s \in [1, \frac{\beta - w\delta_2}{\delta_1}]} PR(s). \tag{73}$$

By using the same logic in the proof of Theorem 1, if the algorithm does not turn on the generator in $s \in [1, \frac{\beta - w\delta_2}{\delta_1}]$, and keeps the generator off, the ratio of the online cost increment over the offline cost increment is

$$R_{off}(\delta_2) = \frac{c_m + \delta_2}{c_m + \frac{Lc_o\alpha}{Lc_o + c_m}\delta_2}. \tag{74}$$

Hence the lower bound can be calculated by finding the minimum of these two values:

$$\text{CR}_l(\delta_1, \delta_2) = \min\{R_{on}(\delta_1, \delta_2), R_{off}(\delta_2)\}. \tag{75}$$

This complete the proof of Lemma 1.

□

Similar to Theorem 1, we want to find (δ_1^*, δ_2^*) that maximizes the lower bound $\text{CR}_l(\delta_1, \delta_2)$. First, for each δ_2, we find a corresponding δ_1 such that $\delta_1 = \arg\max_{\delta} R_{on}(\delta, \delta_2)$. This reduces $R_{on}(\delta_1, \delta_2)$ to a single variable function of δ_2.

$$\delta_1 = \arg\max_{\delta} R_{on}(\delta, \delta_2) \tag{76a}$$

$$\text{s.t.} \quad \delta_2 \leq \delta \leq (\beta - w\delta_2)/w, \tag{76b}$$

$$\delta_2 \leq \delta \leq L(p_{\max} + \eta \cdot c_g - c_o - \frac{c_m}{L}). \tag{76c}$$

For a given δ_2, the function $R_{on}(\delta_1, \delta_2)$ is a concave function of δ_1. Therefore we can easily find a corresponding δ_1 for each δ_2. Now both R_{on} and R_{off} are a function of δ_2, and we find the maximum of the minimum of two single variable functions. We know that for $\delta_2 = 0$, we have $R_{on}(\delta_1, 0) \geq R_{off}(0) = 1$, and $R_{off}(\delta_2)$ is an increasing function. Hence, similar to (18) we keep increasing δ_2 until we find the intersection of the two functions. Therefore, δ_2^*

can be obtained by solving the following optimization problem:

$$\delta_2^* = \arg\max_{\delta_2} CR_l(\delta_1, \delta_2) \tag{77a}$$

$$\text{s.t.} \quad 0 \le \delta_2 \le \beta/(2w), \tag{77b}$$

$$0 \le \delta_2 \le L\left(p_{\max} + \eta \cdot c_g - c_o - \frac{c_m}{L}\right), \tag{77c}$$

$$R_{on}(\delta_1, \delta_2) \ge R_{off}(\delta_2), \tag{77d}$$

$$\delta_1 \text{ is obtained from (76).} \tag{77e}$$

Therefore, we always have $R_{on}(\delta_1^*, \delta_2^*) \ge R_{off}(\delta_2^*)$ and $CR_l(\delta_1^*, \delta_2^*) = R_{off}(\delta_2^*)$, which completes the proof of Theorem 4.

□