Dynamic Provisioning in Next-Generation Data Centers with On-site Power Production

TU, Jinlong

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

The Chinese University of Hong Kong
August 2013

Abstract

The critical need for clean and economical sources of energy is transforming data centers that are primarily energy consumers to also become energy producers. We focus on minimizing the operating costs of next-generation data centers that can jointly optimize the energy supply from on-site generators, renewable sources and the power grid, and the energy demand from servers as well as power conditioning and cooling systems. We formulate the cost minimization problem and present an offline optimal algorithm. For "on-grid" data centers that use the grid and renewable, we devise a deterministic (resp., randomized) online algorithm that achieves a cost of within $2-\alpha_s$ (resp., $e/(e-1+\alpha_s)$) of the offline optimal, where α_s is a normalized look-ahead window size. For "hybrid" data centers that have additional on-site power generators, we develop an online algorithm that achieves a cost of at most $\frac{P_{\max}(2-\alpha_s)}{c_o+c_m/L}\left[1+2\frac{P_{\max}-c_o}{P_{\max}(1+\alpha_g)}\right]$ of the offline optimal, where α_s and α_g are normalized look-ahead window sizes, $P_{\max}$ is the maximum grid power price, and L, c_o, and c_m are parameters of an on-site generator.

Using extensive workload traces from Akamai with the corresponding grid power prices, we simulate our offline and online algorithms in a realistic setting. Our offline (resp., online) algorithm achieves a cost reduction of 13.1% (resp., 7.5%) for an on-grid data center and 26.5% (resp., 20.3%) for a hybrid data center. The cost reductions are quite significant and make a strong case for a joint optimization of energy supply and energy demand in a data center. A hybrid data center provides about 13% additional cost reduction over an on-grid data center representing the additional cost benefits that on-site power generation provides over using the grid and renewable alone.

摘要

对于清洁经济能源的迫切需要，使得主要是能源消耗者的数据中心也成为了能源生产者。本论文专注于最小化下一代数据中心的 经营成本。这些数据中心可以联合优化来自发电机、可再生能源和公共电网的能源，以及来自服务器、功率调节和冷却系统的能 源需求。我们提出了成本最小化问题并且设计了一个离线最优算法。 对于依赖于电网和可再生能源的数据中心，我们设计了一个成本在离线最优$2-\alpha_s$（或$e/(e-1+\alpha_s)$）倍之内的确定性（或随机）在线算法，其中α_s是归一化的前瞻窗口大小。对于有额外发电机的“混合”数据中心，我们设计了一个成本在离线最优$\frac{P_{\max}(2-\alpha_s)}{c_o+c_m/L}\left[1+2\frac{P_{\max}-c_o}{P_{\max}(1+\alpha_g)}\right]$倍之内的在线算法，其中$\alpha_s$和$\alpha_g$是归一化的前瞻窗口大小， $P_{\max}$是电网的最高电价，L、c_o和c_m是发电机的参数。

使用来自Akamai公司的工作量数据以及相应的电网价格，我们在一个真实环境中模拟了离线和在线算法。对于依赖于电网和可再生能源的数据中心，我们的离线（或在线）算法节省了13.1%（或7.5%）的成本；对于有额外发电机的“混合”数据中心，我们的离线（或在线）算法节省了26.5%（或20.3%）的成本。 降低成本是相当显著的，这体现了联合优化数据中心的能源供应和能源需求的重要性。混合数据中心提供了相对于依赖于电网和可再生能源的数据中心约13%的额外成本节约，这些额外的成本节约是由现场发电机提供的。

Acknowledgement

I would like to begin by expressing my deep gratitude to my supervisor, Professor Minghua Chen. He has been a constant source of encouragement and motivation throughout my graduate life. It is his insightful and patient guidance that makes me overcome many difficulties and make progresses on my research topic. I think what I have learned from him is not just the technical knowledge, but his enthusiasm, insights and ways of thinking.

I would like to acknowledge Professor Ramesh K. Sitaraman and Professor Chi-Kin Chau. Professor Ramesh K. Sitaraman has been a research collaborator. He provides valuable comments during my research. His kind help substantially improves the quality of my work. Professor Chi-Kin Chau is also a research collaborator. The clarity of his thoughts and his rigorous academic attitude have influenced me to a great extent.

I have been incredibly lucky to have really great friends throughout my graduate life: Sheng Cai, Zhongqiu Chang, Xiangwen Chen, Hanxu Hou, Tan Jin, Baichuan Li, Ronghua Li, Lian Lu, Tan Lu, Renxin Mao, Ziyu Shao, Jihang Ye, Guanglin Zhang, Jincheng Zhang, Shaoquan Zhang, Bolei Zhou and Zhe Zhu. I would, in particular, like to mention Tan Jin, who has influenced me in many ways, and Lian Lu, who has been a great research collaborator. And I always benefit a lot from discussions with him.

Finally, I would like to mention that all this would not have been possible without supporting from my parents, my brothers and my grandparents.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Contributions 4

1.3 Thesis Organization 8

2 The Data Center Cost Minimization Problem **9**

2.1 Model Assumptions 9

2.2 Problem Formulation 13

2.3 An Offline Optimal Algorithm 17

3 The Benefit of Joint Optimization **19**

3.1 Problem Decomposition 19

3.2 Price of Decomposition 20

4 Online Algorithms for On-grid Data Centers **22**

4.1 Decompose CP into sub-problems CP_is 22

4.2 Deterministic Online Algorithm GCSR 24

4.3 Randomized Online Algorithm RGCSR 27

5 Online Algorithms for Hybrid Data Centers **31**

5.1 A useful structure of an offline optimal solution of EP 31

5.1.1 Decompose EP into sub-problems EP_is 31

5.1.2 Solve each sub-problem EP_i 32

5.2 Online algorithm CHASE 34

5.3 Combining GCSR and CHASE 35

6 Empirical Evaluations **39**

6.1 Parameters and Settings 39

6.2 Impact of Model Parameters on Cost Reduction . . . 42
6.3 The Relative Value of Energy versus Capacity Provisioning . . . 44
6.4 Benefit of Look-ahead . . . 45
6.5 How Much On-site Power Production is Enough . . . 45
6.6 Impact of Renewable Penetration . . . 46
6.7 Randomized Algorithms vs. Deterministic Algorithms . . . 47

7 Conclusion and Future Work **48**

7.1 Conclusion . . . 48
7.2 Discussion . . . 48
7.2.1 The Heterogeneous Scenario . . . 49
7.2.2 A Heuristic Extension of DCMON . . . 50
7.2.3 Empirical Performances . . . 52
7.3 Future Work . . . 54

A Proof of Theorem 2 **55**

B Proofs of Theorems 3, 4 and 5 **59**

B.1 Proof of Theorem 3 . . . 59
B.2 Proof of Theorem 4 . . . 62
B.3 Proof of Theorem 5 . . . 68

C Proofs of Theorems 6, 7 and 8 **70**

C.1 Proof of Theorem 6 . . . 70
C.2 Proof of Theorem 7 . . . 74
C.3 Proof of Theorem 8 . . . 81

D Proof of Theorem 9 **88**

Bibliography **92**

List of Figures

1.1 While an "on-grid" data center derives all its power from the grid, next-generation "hybrid" data centers have additional on-site power generation. 5

4.1 An example of how workload $\boldsymbol{a}$ is decomposed into 4 sub-demands. . 24
4.2 An example of $a_i(t)$ and corresponding solutions obtained by $\mathbf{GCSR_s^{(w)}}$ and $\mathbf{RGCSR_s^{(w)}}$. During the first idle period, the cumulative idling cost does not reach β_s, thus $\mathbf{RGCSR_s^{(0)}}$ and $\mathbf{RGCSR_s^{(w)}}$ keep server idling (*i.e.*, $x_i = 1$). Meanwhile, $\mathbf{RGCSR_s^{(0)}}$ turns off the server when the cumulative idling cost reaches the threshold randomly chosen by it. With looking ahead, $\mathbf{RGCSR_s^{(w)}}$ see the job comes before cumulative idling cost reaches β_s, thus it keeps server idling. During the second idle period, $\mathbf{GCSR_s^{(0)}}$ turns off the server when cumulative idling cost reaches β_s. With looking ahead, $\mathbf{GCSR_s^{(w)}}$ turns off the server w slots earlier than $\mathbf{GCSR_s^{(0)}}$. $\mathbf{RGCSR_s^{(0)}}$ turns off the server when cumulative idling cost reaches the threshold it chooses. $\mathbf{RGCSR_s^{(w)}}$ takes actions w slots earlier than $\mathbf{RGCSR_s^{(0)}}$. 27
4.3 PDF $f_\Lambda(\lambda)$. Here $\alpha_s = 0.4$. 28

5.1 An example of $e_i(t)$, $R_i(t)$ and the corresponding solution obtained by $\mathbf{CHASE_s^{(w)}}$ for $\mathbf{EP_i}$. 33
5.2 The flow chart of **DCMON**. 36
5.3 Theoretical and empirical ratios of algorithm $\mathbf{DCMON^{(w)}}$ vs. look-ahead window size w. 37
5.4 Cumulative cost difference functions (Eqn. 5.3) of three generators in real-world trace based simulations. 38

5.5 Comparison of $x(t), y(t)$ obtained by $\mathbf{DCMON^{(w)}}$ and the offline optimal. 38

6.1 Real-world workload from Akamai and the corresponding grid power prices and renewable energy within the same time period. 40
6.2 Variation of cost reduction with model parameters. 43
6.3 Relative values of CP, EP, and DCM. 44
6.4 Variation of cost reduction with look-ahead and on-site capacity. . . . 46
6.5 Impact of renewable penetration. 47
6.6 Randomized v.s. Deterministic Algorithms. 47

7.1 Performance in the heterogeneous scenario. 54

A.1 An example of $a(t)$, $\bar{x}(t)$, $x^*(t)$, $\bar{y}(t)$ and $y^*(t)$. $\bar{x}(t)$, $\bar{y}(t)$ are obtained by solving **CP** and **EP** sequentially. $x^*(t)$, $y^*(t)$ are obtained by solving **DCM** directly. 58

B.1 Worst case examples. 68

C.1 An example of critical segments. 75

List of Tables

1.1 Summary of algorithmic results. The on-grid results are the best possible. 7

2.1 Key notation. 10

4.1 Summary of the differences between algorithms that we developed in this thesis and previous ones. 23

6.1 Compared Algorithms. 42

7.1 Notation for the heterogeneous scenario. 49

7.2 Parameters of servers and generators. 52

7.3 Parameters of servers and generators. 54

Chapter 1

Introduction

1.1 Motivation

Internet-scale cloud services that deploy large distributed systems of servers around the world are revolutionizing all aspects of human activity. The rapid growth of such services has lead to a significant increase in server deployments in data centers around the world. Energy consumption of data centers accounts for roughly 1.5% of the global energy consumption and is increasing at an alarming rate of about 15% on an annual basis [24]. The surging global energy demand relative to its supply has caused the price of electricity to rise, even while other operating expenses of a data center such as network bandwidth have decreased precipitously. Consequently, the energy costs now represent a large fraction of the operating expenses of a data center today [12], and decreasing the energy expenses has become a central concern for data center operators.

The emergence of energy as a central consideration for enterprises that operate large server farms is drastically altering the traditional boundary between a data center and a power utility (c.f. Fig. 1.1). Traditionally, a data center hosts servers but buys electricity from an utility company through the power grid. However, the criticality of the energy supply is leading data centers to broaden their role to also generate much of the required power on-site, decreasing their dependence on a third-party utility. While data centers have always had generators as a short-term backup for when the grid fails, on-site generators for sustained power supply is a newer trend. For instance, Apple recently announced that it will build a massive data center for its iCloud services with 60% of its energy coming from its on-site generation that uses "clean energy" sources such as fuel cells with biogas and solar panels [28]. As another example, eBay recently announced that it will add a 6 MW facility to its existing data center in Utah that will be largely powered by on-site fuel cell generators [20]. Besides, Bloom Energy Servers (a kind of fuel cell generator) has been installed at various customers including Google, Wal-Mart, AT&T, Staples, The Coca-Cola Company and notable non-profits including Caltech and Kaiser Permanente to provide on-site power [10].

The trend for *hybrid* data centers that generate electricity on-site (c.f. Fig. 1.1) with reduced reliance on the grid is driven by the confluence of several factors. This trend is also mirrored in the broader power industry where the centralized model for power generation with few large power plants is giving way to a more distributed generation model [14] where many smaller on-site generators produce power that is consumed locally over a "microgrid". A key factor favoring on-site generation is the potential for cheaper power than the grid, especially during peak hours. On-site generation also reduces transmission losses that in turn reduces the effective cost, because the power is generated close to where it is consumed. In addition, many enterprises have a mandate to use cleaner renewable energy sources, such as Apple's mandate to use 100% clean energy in its data centers [8]. Such a mandate is more easily achievable with the enterprise generating all or most of its power on-site, especially since recent advances such as the fuel cell technology of Bloom Energy [9] make on-site generation economical and feasible. Finally, the risk of service outages caused by the failure of the grid, as happened recently when thunderstorms brought down the grid causing a denial-of-service for Amazon's AWS service for several hours [21], has provided greater impetus for on-site power generation that can sustain the data center for extended periods without the grid.

In the traditional scenario, the utility is responsible for energy provisioning (**EP**) that has the goal of supplying energy as economically as possible to meet the energy demand, albeit the utility has no detailed knowledge and no control over the server workloads within a data center that drive the consumption of power. The energy provisioning problem takes as input the demand for electricity from the consumers and determines which power generators should be used at what time to satisfy the demand in the most economical fashion. Optimal energy provisioning by the utility in isolation is characterized by the unit commitment problem (UC) [11, 33], including a mixed-integer programming approach [31] approach and a stochastic control approach [37]. All these approaches assume the demand (or its distribution) in the entire horizon is known *a priori*, thus they are applicable only when future input information can be predicted with certain level of accuracy. In contrast, in this thesis we consider an online setting where the algorithms may utilize only information in the current time slot or in the near future (limited look-ahead).

Further, in a traditional scenario, a data center is responsible for capacity provisioning (**CP**) that has the goal of managing its server capacity to serve the incoming workload from end users while reducing the total energy demand of servers [30][13], but without detailed knowledge or control over the power generation. In particular, dynamic provisioning of server capacity by turning off some servers during periods of low workload to reduce the energy demand has been studied in recent years [26][29]. However, none of these two explicitly considers power supply from renewable sources, or power consumption from power conditioning and cooling systems. It is important yet remains open to consider these two factors.

- Renewable energy as a clean and environment friendly energy is widely used in newly built data centers. Yet, renewable energy (*e.g.*, solar and wind energy) often exhibits large uncertainty that adds to the difficulty in designing dynamic provisioning algorithms since it now needs to take into account both the demand uncertainty (caused by workload uncertainty) and supply uncertainty.

- Energy consumed by power conditioning and cooling systems contributes about 38% of the total energy consumption of a typical data center, as a consequence of which it is important to take into account these two systems. From the modeling perspective, power consumption from these two is usually modeled as a convex and increasing function of server utilization [35]. Thus, the data center operating cost optimization problem (including the operating cost of servers, the cooling systems, and the power conditional system) has a convex and increasing objective and integer variables. Designing dynamic provisioning algorithms for problems of this type in face of input uncertainty has not been studied in the literature, including [26] and [29].

This thesis addresses the challenges and designs dynamic provisioning algorithms for minimizing data center operational cost taking into account the renewable supply as well as the power consumption from power conditioning and cooling systems. The algorithms is for solving problems with convex-and-increasing objective function and integer variables, thus filling in a gap between what theory can offer and what is needed in practice.

In addition to the difference of this thesis and existing works in capacity provisioning and energy provisioning, this thesis is also unique in that it jointly optimizes both problems while existing works focus on only one of them. It focuses on the key challenges that arise in the emerging hybrid model for a data center that is able to simultaneously optimize *both* the generation and consumption of energy (c.f. Fig. 1.1). The convergence of power generation and consumption within a single data center entity and the increasing impact of energy costs requires a new integrated approach to both energy provisioning (**EP**) and capacity provisioning (**CP**).

1.2 Contributions

Online vs. Offline Algorithms In designing algorithms for optimizing the operating cost of a hybrid data center, there are four time-varying inputs: the server workload $a(t)$ generated by service requests from users, the price of a unit energy from the grid $p(t)$, the available renewable energy $h(t)$ and the total power consumption function g_t for each time t where $1 \leq t \leq T$. We begin by investigating *offline* algorithms that minimize the operating cost with perfect knowledge of the

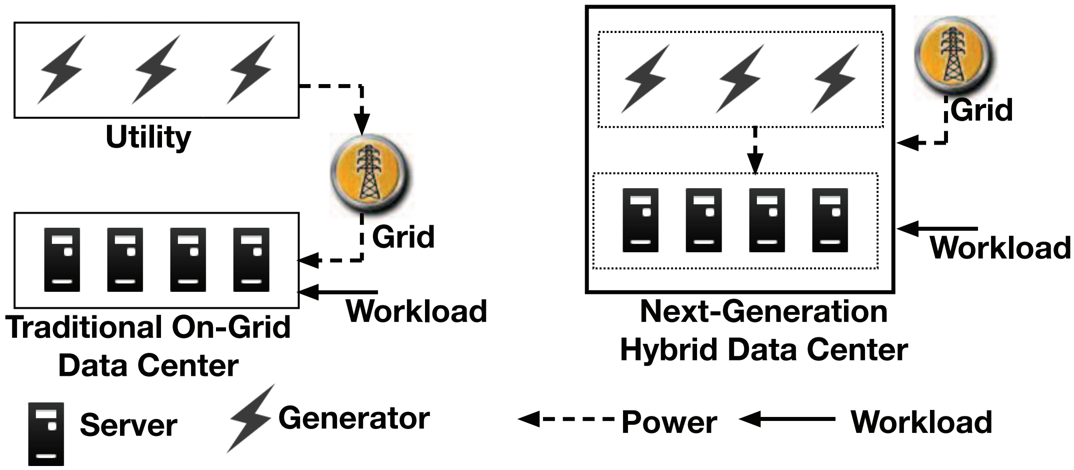


Figure 1.1: While an "on-grid" data center derives all its power from the grid, next-generation "hybrid" data centers have additional on-site power generation.

entire input sequence $a(t)$, $p(t)$, $h(t)$ and g_t, for $1 \leq t \leq T$. However, in real-life, the time-varying input sequences are not knowable in advance. In particular, the optimization must be performed in an *online* fashion where decisions at time t are made with the knowledge of inputs $a(\tau), p(\tau), h(\tau)$ and g_τ, for $1 \leq \tau \leq t+w$, where $w \geq 0$ is a small (possibly zero) look-ahead window. Specifically, an online algorithm has no knowledge of inputs beyond the look-ahead window, *i.e.*, the inputs for time $t + w < \tau \leq T$, are unknown. We assume the inputs within the look-ahead are perfectly known when analyzing the algorithm performance. In practice, short-term demand or grid price can be estimated rather accurately by various techniques including pattern analysis and time series analysis and prediction [22][17]. As is typical in the study of online algorithms [15], we seek theoretical guarantees for our online algorithms by computing the *competitive ratio* that is the maximum ratio (over all possible inputs) between the online algorithm's cost (with no or limited look-ahead) and the offline optimal assuming complete future information. Thus, a small competitive ratio provides a strong guarantee that the online algorithm will achieve a cost close to the offline optimal even for the worst case input.

Our Contributions A key contribution of this thesis is to formulate and study data center cost minimization (**DCM**) that integrates energy procurement from the grid and the renewable sources, energy production using on-site generators, and dynamic server capacity management. Our work jointly optimizes the two components of **DCM**: energy provisioning (**EP**) from the grid and generators and capacity provisioning (**CP**) of the servers. We summarize contributions of this thesis as follows.

- We theoretically evaluate the benefit of joint optimization by showing that optimizing capacity provisioning (**CP**) and energy provisioning (**EP**) sequentially results in a factor loss of optimality $\rho = LP_{\max}/\left(Lc_o + c_m\right)$ compared to optimizing them jointly, where $P_{\max}$ is the maximum grid power price, and L, $c_{o,}$ and c_m are the capacity, incremental cost, and sunk cost of an on-site generator respectively. Further, leveraging the idea of dynamic programming, we derive an efficient offline optimal algorithm for hybrid data centers that jointly optimize **EP** and **CP** to minimize the data center's operating cost.

- For on-grid data centers, we devise a deterministic (resp., randomized) algorithm that achieves a competitive ratio of $2 - \alpha_s$ (resp., $e/(e - 1 + \alpha_s)$), where $\alpha_s \in [0, 1]$ is the normalized look-ahead window size. Further, we show that our algorithm has the best competitive ratio of any deterministic (resp., randomized) online algorithm for the problem (c.f. Tab. 1.1). For hybrid data centers, we devise an online deterministic algorithm that achieves a competitive ratio of $\frac{P_{\max}(2-\alpha_s)}{c_o+c_m/L}\left[1 + 2\frac{P_{\max}-c_o}{P_{\max}(1+\alpha_g)}\right]$, where α_s and α_g are normalized look-ahead window sizes. All the online algorithms perform better as the look-ahead window increases, as they are better able to plan their current actions based on greater knowledge of future inputs. Interestingly, in the on-grid case, we show that there exists a *fixed* threshold value for the look-ahead window for which the online algorithm matches the offline optimal in performance achieving a competitive ratio of 1, *i.e.*, there is no additional benefit gained by the online algorithm if its look-ahead is increased beyond the threshold.

Competitive Ratio	**On-grid**	**Hybrid**
No Look-ahead	2 & $\frac{e}{e-1}$	$\frac{2P_{\max}}{c_o+c_m/L}\left[1+2\frac{P_{\max}-c_o}{P_{\max}}\right]$
With Look-ahead	$2-\alpha_s$ & $\frac{e}{e-1+\alpha_s}$	$\frac{P_{\max}(2-\alpha_s)}{c_o+c_m/L}\left[1+2\frac{P_{\max}-c_o}{P_{\max}(1+\alpha_g)}\right]$

Table 1.1: Summary of algorithmic results. The on-grid results are the best possible.

- Using extensive workload traces from Akamai and the corresponding grid prices and renewable energy supply, we simulate our offline and online algorithms in a realistic setting with the goal of empirically evaluating their performance. Our offline (resp., online) algorithm achieves a cost reduction of 13.1% (resp., 7.5%) for an on-grid data center and 26.5% (resp., 20.3%) for a hybrid data center. The cost reduction is computed in comparison with the baseline cost achieved by the current practice of statically provisioning the servers and using the power grid and the renewable energy. The cost reductions are quite significant and make a strong case for utilizing our joint cost optimization framework. Furthermore, our online algorithms obtain almost the same cost reduction as the offline optimal solution even with a small look-ahead of 6 hours, indicating the value of short-term prediction of inputs.

- A hybrid data center provides about 13% additional cost reduction over an on-grid data center representing the additional cost benefits that on-site power generators provide over using the grid and renewable energy. Interestingly, it is sufficient to deploy a partial on-site generation capacity that provides 60% of the peak power requirements of the data center to obtain over 95% of the additional cost reduction. This provides strong motivation for a traditional on-grid data center to deploy at least a partial on-site generation capability to save costs.

1.3 Thesis Organization

This thesis is organized as follows. In Chapter 2, we formulate the data center cost minimization problem and present a dynamic programming based offline optimal algorithm. Chapter 3 discusses the benefit of joint optimization over optimizing capacity provisioning and energy provisioning sequentially. In Chapter 4, we present our deterministic and randomized algorithms for on-grid data centers. Chapter 5 discusses how to design online algorithms for hybrid data centers. In Chapter 6, we carry out extensive real-world trace based simulations to demonstrate the effectiveness of our algorithms in practice. Chapter 7 concludes the thesis with discussions and future work.

□ **End of chapter.**

Chapter 2

The Data Center Cost Minimization Problem

We consider the scenario where a data center can jointly optimize energy production, procurement, and consumption so as to minimize its operating expenses. We refer to this data center cost minimization problem as **DCM**. To study **DCM**, we model how energy is produced using on-site power generators and renewable sources, how it can be procured from the power grid, and how data center capacity can be provisioned dynamically in response to workload. While some of these aspects have been studied independently, our work is unique in optimizing these dimensions simultaneously as next-generation data centers can. Our algorithms minimize cost by use of techniques such as: (i) dynamic capacity provisioning of servers – turning off unnecessary servers when workload is low to reduce the net energy consumption (ii) opportunistic energy procurement – switching between the on-site and grid energy sources to exploit price fluctuation, and (iii) dynamic provisioning of generators – orchestrating which generators produce what portion of the energy demand. While prior literature has considered these techniques in isolation, we show how they can be used in coordination to manage both the supply and demand of power to achieve substantial cost reduction.

2.1 Model Assumptions

We adopt a discrete-time model whose time slot matches the timescale at which scheduling decisions are performed.

Notation	**Definition**
T	Number of time slots
N	Number of on-site generators
β_s	Switching cost of a server ($)
β_g	Startup cost of an on-site generator ($)
c_m	Sunk cost of maintaining a generator in its active state per slot ($)
c_o	Incremental cost for an active generator to output an additional unit of energy ($/Wh)
L	The maximum output of a generator (Watt)
$a(t)$	Workload at time t
$p(t)$	Price per unit energy drawn from the grid at t ($P_{\min} \le p(t) \le P_{\max}$) ($/Wh)
$h(t)$	Renewable energy at time t
$x(t)$	Number of active servers at t
$v(t)$	Grid power used at t (Watt)
$y(t)$	Number of active on-site generators at t
$u(t)$	Total power output from active generators at t (Watt)
$g_t(x(t), a(t))$	Total power consumption as a function of $x(t)$ and $a(t)$ at t (Watt)

Note: we use bold symbols to denote vectors, *e.g.*, $\boldsymbol{x} = \langle x(t) \rangle$. Brackets indicate the unit.

Table 2.1: Key notation.

Workload model. Similar to existing work [16, 36, 19], we consider a "mice" type of workload for the data center where each job has a small transaction size and short duration. Jobs arriving in a slot get served in the same slot. Workload can be split among active servers at arbitrary granularity like a fluid. These assumptions model a "request-response" type of workload that characterizes serving web content or hosted application services that entail short but real-time interactions between the user and the server. Without loss of generality, we assume there are totally T slots, and each has a unit length. The workload to be served at time t is represented by $a(t)$. Note that we do not make any stochastic assumption about $a(t)$.

Server model. We assume that the data center consists of a sufficient number of homogeneous servers, and each has unit service capacity, *i.e.*, it can serve at most one unit workload per slot. Let $x(t)$ be the number of active servers. We model the aggregate server power consumption as $b(t) \triangleq f_s(x(t), a(t))$, an increasing (*i.e.*, non-decreasing) and convex function of $x(t)$ and $a(t)$. That is, the first and second order partial derivatives in $x(t)$ and $a(t)$ are all non-negative. In addition, to get the workload served in the same slot, we must have $x(t) \geq a(t)$.

This power consumption model is quite general and captures many common server models. One example is the commonly adopted standard linear model [12]:

$$f_s(x(t), a(t)) = c_{idle} x(t) + (c_{peak} - c_{idle}) a(t),$$

where c_{idle} and c_{peak} are the power consumed by an server at idle and fully utilized state, respectively. Most servers today consume significant amounts of power even when idle. A holy grail for server design is to make them "power proportional" by making c_{idle} zero [34].

Besides power costs, turning a server on entails switching cost [30], denoted as β_s, including the amortized service interruption cost, wear-and-tear cost, *e.g.*, component procurement, replacement cost (hard-disks in particular) and risk associated with server switching. It is comparable to the energy cost of running a server for several hours [26].

In addition to servers, power conditioning and cooling systems also consume a significant portion of power. The three contribute about 94% of overall power consumption and their power draw vary drastically with server utilization [35]. Thus, it is important to model the power consumed by power conditioning and cooling systems.

Power conditioning system model. Power conditioning system usually includes power distribution units (PDUs) and uninterruptible power supplies (UPSs).

PDUs transform the high voltage power distributed throughout the data center to voltage levels appropriate for servers. UPSs provides temporary power during outage. We model the power consumption of this system as $f_p(b(t))$, an increasing and convex function of the aggregate server power consumption $b(t)$.

This model is general and one example is a quadratic function adopted in a comprehensive study on the data center power consumption [35]: $f_p(b(t)) = C_1 + \pi_1 b^2(t)$, where $C_1 > 0$ and $\pi_1 > 0$ are constants depending on specific PDUs and UPSs.

Cooling system model. We model the power consumed by the cooling system as $f_c^t(b(t))$, a time-dependent (*e.g.*, depends on ambient weather conditions) increasing and convex function of $b(t)$.

This cooling model captures many common cooling systems. According to [27], the power consumption of an outside air cooling system can be modelled as a time-dependent cubic function of $b(t)$: $f_c^t(b(t)) = K_t b^3(t)$, where $K_t > 0$ depends on ambient weather conditions (*e.g.*, air temperature) at time t. According to [35], the power draw of a water chiller cooling system can be modelled as a time-dependent quadratic function of $b(t)$: $f_c^t(b(t)) = Q_t b^2(t) + L_t b(t) + C_t$, where $Q_t, L_t, C_t \geq 0$ depend on outside air and chilled water temperature at time t.

On-site generator model. We assume that the data center has N units of homogeneous on-site generators, each having a maximum output of L. Similar to generator models studied in the unit commitment problem [23], we define a generator startup cost β_g, which typically involves heating up cost, additional maintenance cost due to each startup (*e.g.*, fatigue and possible permanent damage resulted by stresses during startups), c_m as the sunk cost of maintaining a generator in its active state for a slot, and c_o as the incremental cost for an active generator to output an additional unit of energy. Thus, the total cost for $y(t)$ active generators that output $u(t)$ units of energy at time t is $c_m y(t) + c_o u(t)$.

Renewable energy model. We assume that the data center is also equipped with renewable energy sources, such as solar panels and wind farms. At each time

t, the renewable sources generate an arbitrary $h(t)$ amount of energy available for use. Note that we do not make any stochastic assumption about $h(t)$. It is clear that the data center should use the renewable energy to satisfy its demand first before seeking alternative supply since the renewable energy is clean and its price is zero. The available renewable energy is usually not enough to satisfy the entire data center energy demand at any given time. However, if it exceeds the energy requirements of the data center, it could be stored for later use. We do not consider energy storage in our current work and is an interesting direction for future research.

Grid model. The grid supplies energy to the data center in an "on-demand" fashion, with time-varying price $p(t)$ per unit energy at time t, where $0 \leq P_{\min} \leq p(t) \leq P_{\max}$. Thus, the cost of drawing $v(t)$ units of energy from the grid at time t is $p(t)v(t)$.

To keep the study interesting and practically relevant, we make the following assumptions: (i) the server and generator turning-on cost are strictly positive, *i.e.*, $\beta_s > 0$ and $\beta_g > 0$. (ii) the cost of generating L units of energy using a single generator, which is $c_o L + c_m$, is less than the maximum cost of buying the same L units of power from the grid at a cost of $P_{\max} L$, *i.e.*, $c_o + c_m/L < P_{\max}$. Otherwise, it should be clear that it is optimal to always buy energy from the grid, because in that case the grid energy is always cheaper and incurs no startup costs.

2.2 Problem Formulation

Based on the above models, the total power consumption of the data center is the sum of the power consumed by servers, power conditioning system and cooling system. We express the total power consumption as a time-dependent function

$$g_t(x(t), a(t)) \triangleq b(t) + f_p(b(t)) + f_c^t(b(t)),$$

where $b(t) = f_s(x(t), a(t))$ is the power consumed by servers, and $f_p(b(t))$ and $f_c^t(b(t))$ represent power consumed by power conditioning and cooling systems, respectively. We remark that $g_t(x(t), a(t))$ is increasing and convex in $x(t)$ and $a(t)$, because it is the sum of three increasing and convex functions. *Note that all results we derive in this thesis apply to any $g_t(x, a)$ as long as it is increasing (i.e., non-decreasing) and convex in x and a.*

Our objective is to minimize the total cost of the data center in entire horizon $[1, T]$, which is given by

$$\begin{aligned}\text{Cost}(x, y, u, v) \triangleq \sum_{t=1}^{T} \{ & v(t)p(t) + c_o u(t) + c_m y(t) \\ & + \beta_s [x(t) - x(t-1)]^+ + \beta_g [y(t) - y(t-1)]^+ \} ,\end{aligned} \tag{2.1}$$

which includes the cost of grid electricity, the running cost of on-site generators, and the switching cost of servers and on-site generators in the entire horizon $[1, T]$. Throughout this thesis, we set our initial condition to be $x(0) = y(0) = 0$.

We formally define the data center cost minimization problem as a non-linear mixed-integer program, given the workload $a(t)$, the grid price $p(t)$, the renewable energy $h(t)$ and the time-dependent function $g_t(x, a)$, for $1 \le t \le T$, as time-varying inputs.

$$\min_{x,y,u,v} \quad \text{Cost}(x, y, u, v) \tag{2.2}$$

$$\text{s.t.} \quad u(t) + v(t) + h(t) \ge g_t(x(t), a(t)), \tag{2.3}$$

$$u(t) \le L y(t), \tag{2.4}$$

$$x(t) \ge a(t), \tag{2.5}$$

$$y(t) \le N, \tag{2.6}$$

$$x(0) = y(0) = 0, \tag{2.7}$$

$$\text{var} \quad x(t), y(t) \in \mathbb{N}^0, u(t), v(t) \in \mathbb{R}_0^+, \; t \in [1, T],$$

where $[\cdot]^+ = \max(0, \cdot)$, and $\mathbb{N}^0$ and $\mathbb{R}_0^+$ represent the set of non-negative integers and real numbers, respectively.

Constraint (2.3) ensures the total power consumed by the data center is jointly supplied by the generators, the grid and the renewable sources. Constraint (2.4) captures the maximal output of the on-site generator. Constraint (2.5) specifies that there are enough active servers to serve the workload. Constraint (2.6) ensures the number of active generators is at most N. Constraint (2.7) is the initial condition.

Note that this problem is challenging to solve. First, it is a non-linear mixed-integer optimization problem. Further, the objective function values across different slots are correlated via the switching costs $\beta_s[x(t)-x(t-1)]^+$ and $\beta_g[y(t)-y(t-1)]^+$, and thus cannot be decomposed. Finally, in practice we need to solve the problem in an online fashion; more specifically, at each time, we don't know the inputs beyond the look-ahead window.

Next, we introduce a proposition to simplify the structure of the problem. Note that if $(x(t))_{t=1}^T$ and $(y(t))_{t=1}^T$ are given, the problem in (2.2)-(2.7) reduces to a linear program and can be solved independently for each slot. We then obtain the following.

Proposition 1. *Given any $x(t)$ and $y(t)$, the $u(t)$ and $v(t)$ that minimize the cost in (2.2) with any $g_t(x,a)$ that is increasing in x and a, are given by: $\forall t \in [1,T]$,*

$$u(t) = \begin{cases} 0, & \text{if } p(t) \le c_o, \\ \min\left(Ly(t), [g_t(x(t),a(t)) - h(t)]^+\right), & \text{otherwise,} \end{cases}$$

and

$$v(t) = [g_t(x(t),a(t)) - h(t)]^+ - u(t).$$

Note that $u(t), v(t)$ can be computed using *only* $x(t), y(t)$ at current time t, thus can be determined in an online fashion. For the ease of presentation, denote $d_t(x(t)) = [g_t(x(t),a(t)) - h(t)]^+$, which is also increasing and convex in $x(t)$. $d_t(x(t))$ represents the residual power demand after using the renewable energy available at time t[1].

[1] In case the renewable supply is abundant and exceeds the entire data center energy demand at time t, the unused renewable is discarded. In practice, it is conceivable that the data center can store the unused renewable into energy storage devices for later use. Our current model does not incorporate energy storage devices and it is a limitation of our current study. Extending our study beyond the limitation is an interesting future direction.

Intuitively, the above proposition says that we should first use the renewable energy to satisfy the overall electricity demand. Then for the residual demand, if the on-site energy price c_o is higher than the grid price $p(t)$, we should buy energy from the grid; otherwise, it is the best to buy the cheap on-site energy up to its maximum supply $L \cdot y(t)$ and the rest (if any) from the more expensive grid. With the above proposition, we can reduce the non-linear mixed-integer program in (2.2)-(2.7) with variables $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{u}$, and $\boldsymbol{v}$ to the following integer program with only variables $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$
\begin{aligned}
& \textbf{DCM}: \\
\min \quad & \sum_{t=1}^{T} \left\{ \psi\left(y(t), p(t), d_t(x(t))\right) + \beta_s [x(t) - x(t-1)]^+ \right. \\
& \left. + \beta_g [y(t) - y(t-1)]^+ \right\} \qquad (2.8) \\
\text{s.t.} \quad & x(t) \geq a(t), \\
& (2.6), (2.7), \\
\text{var} \quad & x(t), y(t) \in \mathbb{N}^0,\ t \in [1, T],
\end{aligned}
$$

where $\psi\left(y(t), p(t), d_t(x(t))\right)$ replaces the term $v(t)p(t) + c_o u(t) + c_m y(t)$ in the original cost function in (2.2) and is defined as

$$
\begin{aligned}
& \psi\left(y(t), p(t), d_t(x(t))\right) \qquad (2.9) \\
\triangleq & \begin{cases}
c_m y(t) + p(t) d_t(x(t)), & \text{if } p(t) \leq c_o, \\
c_m y(t) + c_o L y(t) + & \text{if } p(t) > c_o \text{ and} \\
p(t)\left(d_t(x(t)) - L y(t)\right), & d_t(x(t)) > L y(t), \\
c_m y(t) + c_o d_t(x(t)), & \text{else.}
\end{cases}
\end{aligned}
$$

As a result of the analysis above, it suffices to solve the above formulation of **DCM** with only variables $\boldsymbol{x}$ and $\boldsymbol{y}$, in order to minimize the data center operating cost.

2.3 An Offline Optimal Algorithm

We present an offline optimal algorithm for solving problem **DCM** using Dijkstra's shortest path algorithm [18]. We construct a graph $G = (V, E)$, where each vertex denoted by the tuple $\langle x, y, t\rangle$ represents a state of the data center where there are x active servers, and y active generators at time t. We draw a directed edge from each vertex $\langle x(t-1), y(t-1), t-1\rangle$ to each possible vertex $\langle x(t), y(t), t\rangle$ to represent the fact that the data center can transition from the first state to the second state. Further, we associate the cost of that transition shown below as the weight of the edge:

$$\psi\left(y(t), p(t), d_t(x(t))\right) + \beta_s[x(t) - x(t-1)]^+$$
$$+\beta_g[y(t) - y(t-1)]^+.$$

Next, we find the minimum weighted path from the initial state represented by vertex $\langle 0, 0, 0\rangle$ to the final state represented by vertex $\langle 0, 0, T+1\rangle$ by running Dijkstra's algorithm on graph G. Since the weights represent the transition costs, it is clear that finding the minimum weighted path in G is equivalent to minimizing the total transitional costs. Thus, our offline algorithm provides an optimal solution for problem **DCM**.

Theorem 1. *The algorithm described above finds an optimal solution to problem **D-CM** in time $O\left(M^2N^2T\log\left(MNT\right)\right)$, where T is the number of slots, N the number of generators and $M = \max_{1\le t\le T}\lceil a(t)\rceil$.*

Proof. Since the numbers of active servers and generators are at most M and N, respectively, and there are $T+2$ time slots, graph G has $O(MNT)$ vertices and $O(M^2N^2T)$ edges. Thus, the run time of Dijkstra's algorithm on graph G is $O\left(M^2N^2T\log\left(MNT\right)\right)$. □

Remark: In practice, the time-varying input sequences ($p(t)$, $a(t)$, $h(t)$ and g_t) may not be available in advance and hence it may be difficult to apply the above offline algorithm. However, an offline optimal algorithm can serve as a benchmark, using which we can evaluate the performance of online algorithms.

□ **End of chapter.**

Chapter 3

The Benefit of Joint Optimization

Data center cost minimization (**DCM**) entails the joint optimization of both server capacity that determines the energy demand and on-site power generation that determines the energy supply. Now consider the situation where the data center optimizes the net energy demand and supply separately.

3.1 Problem Decomposition

First, the data center dynamically provisions the server capacity according to the grid power price $p(t)$ and the net power consumption function $d_t(x)$ (recall that $d_t(x) = [g_t(x, a(t)) - h(t)]^+$). More formally, it solves the *capacity provisioning* problem which we refer to as **CP** below.

$$
\begin{aligned}
\mathbf{CP}: \quad & \min && \sum_{t=1}^{T} \left\{ p(t) \cdot d_t(x(t)) + \beta_s [x(t) - x(t-1)]^+ \right\} \\
& \text{s.t.} && x(t) \geq a(t), \\
& && x(0) = 0, \\
& \text{var} && x(t) \in \mathbb{N}^0,\ t \in [1, T].
\end{aligned}
$$

Solving problem **CP** yields $\bar{\boldsymbol{x}}$. Thus, the total net power demand at time t given $\bar{x}(t)$ is $d_t(\bar{x}(t))$. Note that $d_t(\bar{x}(t))$ is residual power demand of servers, power conditioning and cooling systems after using the renewable energy at time t.

Second, the data center minimizes the cost of satisfying the net power demand due to $d_t(\bar{x}(t))$, using both the grid and the on-site generators. Specifically, it solves the *energy provisioning* problem which we refer to as **EP** below.

$$\begin{aligned}
\textbf{EP}: \quad \min \quad & \sum_{t=1}^{T} \left\{ \psi\left(y(t), p(t), d_t(\bar{x}(t))\right) + \beta_g [y(t) - y(t-1)]^+ \right\} \\
& y(0) = 0, \\
\text{var} \quad & y(t) \in \mathbb{N}^0,\ t \in [1, T].
\end{aligned}$$

3.2 Price of Decomposition

Let $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ be the solution obtained by solving **CP** and **EP** separately in sequence and $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be the solution obtained by solving the joint-optimization **DCM**. Further, let $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}, \boldsymbol{y})$ be the value of the data center's total cost for solution $(\boldsymbol{x}, \boldsymbol{y})$, including both generator and server costs as represented by the objective function (2.8) of problem **DCM**. The additional benefit of joint optimization over optimizing independently is simply the relationship between $\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ and $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*)$. It is clear that $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ obeys all the constraints of **DCM** and hence is a feasible solution of **DCM**. Thus, $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*) \leq \mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$. We can measure the factor loss in optimality ρ due to optimizing separately as opposed to optimizing jointly on the worst-case input as follows:

$$\rho \triangleq \max_{\text{all inputs}} \frac{\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})}{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*)}.$$

The following theorem characterizes the benefit of joint optimization over optimizing independently.

Theorem 2. *The factor loss in optimality ρ by solving the problem **CP** and **EP** in sequence as opposed to optimizing jointly is given by $\rho = LP_{\max}/\left(Lc_o + c_m\right)$ and it is tight.*

Proof. Refer to Appendix A. □

The above theorem guarantees that for *any* time duration T, *any* workload $\boldsymbol{a}$, *any* grid price $\boldsymbol{p}$, *any* renewable energy $\boldsymbol{h}$ and *any* power consumption function

$g_t(x,a)$ as long as it is increasing and convex in x and a, solving problem **DCM** by first solving **CP** then solving **EP** in sequence yields a solution that is within a factor $LP_{\max}/\left(Lc_o + c_m\right)$ of solving **DCM** directly. Further, the ratio is tight in that there exists an input to **DCM** where the ratio $\mathrm{C_{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})/\mathrm{C_{DCM}}\left(\boldsymbol{x}^*, \boldsymbol{y}^*\right)$ equals $LP_{\max}/\left(Lc_o + c_m\right)$.

The theorem shows in a quantitative way that a larger price discrepancy between the maximum grid price and the on-site power yields a larger gain by optimizing the energy provisioning and capacity provisioning jointly. Over the past decade, utilities have been exposing a greater level of grid price variation to their customers with mechanisms such as time-of-use pricing where grid prices are much more expensive during peak hours than during the off-peak periods. This likely leads to larger price discrepancy between the grid and the on-site power. In that case, our result implies that a joint optimization of power and server resources is likely to yield more benefits to a hybrid data center.

Besides characterizing the benefit of jointly optimizing power and server resources, the decomposition of problem **DCM** into problems **CP** and **EP** provides a key approach for our online algorithm design. Problem **DCM** has an objective function with mutually-dependent coupled variables $\boldsymbol{x}$ and $\boldsymbol{y}$ indicating the server and generator states, respectively. This coupling as expressed in function $\psi\left(y(t), p(t), d_t(x(t))\right)$ makes it difficult to design provably good online algorithms. However, instead of solving problem **DCM** directly, we devise online algorithms to solve problems **CP** that involves only server variable $\boldsymbol{x}$ and **EP** that involves only the generator variables $\boldsymbol{y}$. Combining the online algorithms for **CP** and **EP** respectively yields the desired online algorithm for **DCM**.

□ **End of chapter.**

Chapter 4

Online Algorithms for On-grid Data Centers

We first develop online algorithms for **DCM** for an *on-grid* data center, where there are grid power and renewable sources, but no on-site power generators. This captures most data centers operational scenarios today. Since an *on-grid* data center has no on-site power generators, solving **DCM** for it reduces to solving problem **CP** described in Sec. 3.1.

Problems of this kind have been studied in the literature (see *e.g.*, [26][29]). The differences of our work from [26][29] are as follows (also summarized in Tab. 4.1). From the modelling aspect, we explicitly take into account power consumption of both cooling and power conditioning systems, in addition to servers, and power supply from renewable energy sources. From the formulation aspect, we are solving a different optimization problem, *i.e.*, an integer program with convex and increasing objective function. From the theoretical result aspect, we develop a deterministic online algorithm with competitive ratio $2 - \alpha_s$ and a randomized online algorithm with competitive ratio $e/(e-1+\alpha_s)$, where α_s is the normalized look-ahead window size. Both ratios quickly decrease to 1 as look-ahead window w increase.

4.1 Decompose CP into sub-problems CP_is

Recall that **CP** takes as input the workload $\boldsymbol{a}$, the grid price $\boldsymbol{p}$, the renewable energy $\boldsymbol{h}$ and the time-dependent function g_t, $\forall t$ and outputs the number of active servers $\boldsymbol{x}$. We construct solutions to **CP** in a divide-and-conquer fashion. We first

	Pow. cond., Cooling Renewable	Optimization Problem		C.R.	
		Objective Function	Variable Type	Deter. Alg.	Rand. Alg.
LCP [26]	No	convex	continuous	3	$\times$
CSR/RCSR [29]	No	linear	integer	$2-\alpha_s$	$\frac{e}{e-1+\alpha_s}$
GCSR/RGCSR this thesis	Yes	convex & increasing	integer	$2-\alpha_s$	$\frac{e}{e-1+\alpha_s}$

Note: Here Pow. Cond. stands for Power Conditioning. C.R. stands for Competitive Ratio. Deter. stands for Deterministic. Rand. stands for randomized. Alg. stands for Algorithm. α_s is the normalized look-ahead window size, whose representations are different under the different settings of [29] and this thesis.

Table 4.1: Summary of the differences between algorithms that we developed in this thesis and previous ones.

decompose the demand $\boldsymbol{a}$ into sub-demands, define a sub-problem associated with each sub-demand for each server, and then solve capacity provisioning *separately* for each sub-problem. Note that the key is to correctly decompose the demand and define the subproblems so that the combined solution is still optimal. More specifically, we slice the demand as follows: for $1 \leq i \leq M = \max_{1\leq t\leq T} \lceil a(t) \rceil$, $1 \leq t \leq T$,

$$a_i(t) \triangleq \min\{1, \max\{0, a(t)-(i-1)\}\}. \tag{4.1}$$

And the corresponding sub-problem $\mathbf{CP}_{\mathrm{i}}$ is defined as follows.

$$\begin{aligned}
\mathbf{CP}_{\mathrm{i}}: \quad \min \quad & \sum_{t=1}^{T} \left\{ p(t) \cdot d_t^i \cdot x_i(t) + \beta_s [x_i(t) - x_i(t-1)]^+ \right\} \\
\text{s.t.} \quad & x_i(t) \geq a_i(t), \\
& x_i(0) = 0, \\
\text{var} \quad & x_i(t) \in \{0,1\},\ t \in [1,T],
\end{aligned}$$

where $x_i(t)$ indicates whether the i-th server is on at time t and $d_t^i \triangleq d_t(i) - d_t(i-1)$. d_t^i can be interpreted as the power consumption increment due to using the i-th

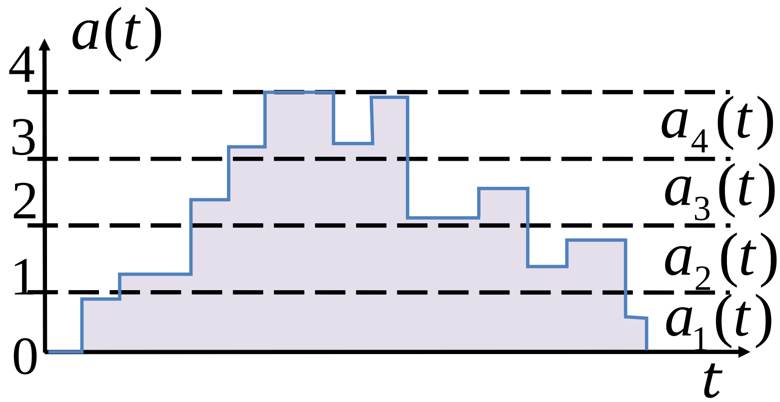


Figure 4.1: An example of how workload $\boldsymbol{a}$ is decomposed into 4 sub-demands.

server at t. Problem $\mathbf{CP}_\mathrm{i}$ solves the capacity provisioning problem with inputs workload $\boldsymbol{a_i}$, grid price $\boldsymbol{p}$ and d_t^i. The key reason for our decomposition is that $\mathbf{CP}_\mathrm{i}$ is easier to solve, since $\boldsymbol{a_i}$ take values in $[0,1]$ and exactly one server is required to serve each $\boldsymbol{a_i}$. As the following theorem states, the individual optimal solutions for problems $\mathbf{CP}_\mathrm{i}$ can be put together to form an optimal solution to the original problem $\mathbf{CP}$. Denote $\mathrm{C}_{\mathrm{CP_i}}(\boldsymbol{x_i})$ as the cost of solution $\boldsymbol{x_i}$ for problem $\mathbf{CP}_\mathrm{i}$ and $\mathrm{C}_{\mathrm{CP}}(\boldsymbol{x})$ the cost of solution $\boldsymbol{x}$ for problem $\mathbf{CP}$.

Theorem 3. *Consider problem* $\mathbf{CP}$ *with any* $d_t(x)$ *that is convex in* x*. Let* $\bar{\boldsymbol{x}}_i$ *be an optimal solution and* $\boldsymbol{x}_i^{on}$ *an online solution for problem* $\mathbf{CP_i}$ *with workload* $\boldsymbol{a}_i$*, then* $\sum_{i=1}^M \bar{\boldsymbol{x}}_i$ *is an optimal solution for* $\mathbf{CP}$ *with workload* $\boldsymbol{a}$*. Furthermore, if* $\forall \boldsymbol{a}_i, i$*, we have* $\mathrm{C}_{\mathrm{CP_i}}(\boldsymbol{x}_i^{on}) \leq \gamma \cdot \mathrm{C}_{\mathrm{CP_i}}(\bar{\boldsymbol{x}}_i)$ *for a constant* $\gamma \geq 1$*, then* $\mathrm{C}_{\mathrm{CP}}(\sum_{i=1}^M \boldsymbol{x}_i^{on}) \leq \gamma \cdot \mathrm{C}_{\mathrm{CP}}(\sum_{i=1}^M \bar{\boldsymbol{x}}_i)$*,* $\forall \boldsymbol{a}$*.*

Proof. Refer to Appendix B.1. □

4.2 Deterministic Online Algorithm GCSR

According to Theorem 3, it remains to design algorithms for each $\mathbf{CP}_\mathrm{i}$. To solve $\mathbf{CP}_\mathrm{i}$ in an online fashion one need only orchestrate one server to satisfy the workload $\boldsymbol{a}_i$ and minimize the total cost. When $a_i(t) > 0$, we must keep the server active to satisfy the workload. The challenging part is what we should do if the server is already active but $a_i(t) = 0$. Should we turn off the server immediately or keep it idling for some time? Should we distinguish the scenarios when the grid price is high versus low?

Inspired by "ski-rental" [15] and [29], we solve $\mathbf{CP}_{\rm i}$ using the following "break-even" idea. During an idle period, *i.e.*, $a_i(t) = 0$, we accumulate an "idling cost" ($\sum_{\tau=t'}^{t} p(\tau)d_\tau^i$, where t' is the first slot of the idle period and t is the current slot) and when it reaches β_s, we turn off the server; otherwise, we keep the server idling. Specifically, our online algorithm $\mathbf{GCSR_s^{(w)}}$ (Generalized Collective Server Rental) for $\mathbf{CP}_{\rm i}$ has a look-ahead window w. At time t, if there exist $\tau' \in [t, t+w]$ such that the idling cost till τ' is at least β_s, we turn off the server; otherwise, we keep it idling. An illustration of $\mathbf{GCSR_s^{(w)}}$ is shown in Fig. 4.2.

Algorithm 1 $\mathbf{GCSR^{(w)}}$ for problem $\mathbf{CP}$

```
initial condition: C_i = 0, x_i(0) = 0, ∀i ∈ [1, M]
at current time t, do
decompose (a(τ))_{τ=t}^{t+w} into (a_1(τ), a_2(τ), ..., a_M(τ))_{τ=t}^{t+w} according to Eqn. (4.1)
  and obtain (d_τ^i)_{τ=t}^{t+w}, ∀i ∈ [1, M]
for i = 1 to M do
    set τ' ← min{t' ∈ [t, t+w] | C_i + Σ_{τ=t}^{t'} p(τ) d_τ^i ≥ β_s}
    if a_i(t) > 0 then
        x_i(t) = 1 and C_i = 0
    else if τ' = NULL or ∃τ ∈ [t, τ'], a_i(τ) > 0 then
        x_i(t) = x_i(t−1) and C_i = C_i + p(t) d_t^i x_i(t)
    else
        x_i(t) = 0 and C_i = 0
    end if
end for
x(t) = Σ_{i=1}^{M} x_i(t)
```

Our online algorithm for **CP**, denoted as $\mathbf{GCSR^{(w)}}$, first employs $\mathbf{GCSR_s^{(w)}}$ to solve each $\mathbf{CP}_\mathrm{i}$ on workload $\boldsymbol{a}_i$, $1 \le i \le M$, in an online fashion to produce output $\boldsymbol{x}_i^{on}$ and then simply outputs $\sum_{i=1}^{M} \boldsymbol{x}_i^{on} = \boldsymbol{x}^{on}$ as the output for the original problem **CP**. More formally, we have Algorithm 1. The competitive analysis of $\mathbf{GCSR^{(w)}}$ is in Theorem 4.

Theorem 4. $\mathbf{GCSR^{(w)}}$ *achieves a competitive ratio of* $2 - \alpha_s$ *for* **CP**, *where* $\alpha_s \triangleq \min(1, wd_{\min}P_{\min}/\beta_s) \in [0, 1]$ *is a "normalized" look-ahead window size and* $d_{\min} \triangleq \min_t\{d_t(1) - d_t(0)\}$. *Further, no deterministic online algorithm with a look-ahead window* w *can achieve a smaller competitive ratio.*

Proof. Refer to Appendix B.2. □

If $d_t(x)$ is strictly increasing in x, we must have $d_{\min} > 0$. In this case, the competitive ratio decreases as the look-ahead window size w increases. Further, when the look-ahead window size w reaches a break-even interval $\Delta_s \triangleq \beta_s/(d_{\min}P_{\min})$, $\mathbf{GCSR^{(w)}}$ has a competitive ratio of 1. That is, having a look-ahead window larger than Δ_s will not decrease the cost any further. Intuitively, this is because when w is no less than Δ_s, $\mathbf{GCSR^{(w)}}$ can tell whether the cumulative idling cost will reach β_s at the beginning of an idle period, thus it will perform exactly the same as the offline optimal.

If $d_t(x)$ is not strictly increasing in x (*e.g.*, there is enough renewable energy to satisfy the entire data center energy demand), then $d_{\min}$ may be 0. In this case, the competitive ratio of $\mathbf{GCSR^{(w)}}$ does not decrease as w increases (remains 2). However, the competitive ratio 2 is for the worst case inputs. In practice, $\mathbf{GCSR^{(w)}}$ can still benefit from looking ahead. As long as power consumption $g_t(x, a)$ is not always less than renewable energy $h(t)$, there exists an index i such that d_t^i is not 0. Then, for the i-th server, with look-ahead, $\mathbf{GCSR^{(w)}}$ can tell whether the cumulative idling cost will reach β_s earlier than without look-ahead. In this way, it benefits from future information.

In addition, according to the realistic setting in Chapter 6, we observe that

$g_t(x, a)$ is strictly increasing in x and is always larger than the renewable $h(t)$. Consequently, $d_{\min} > 0$.

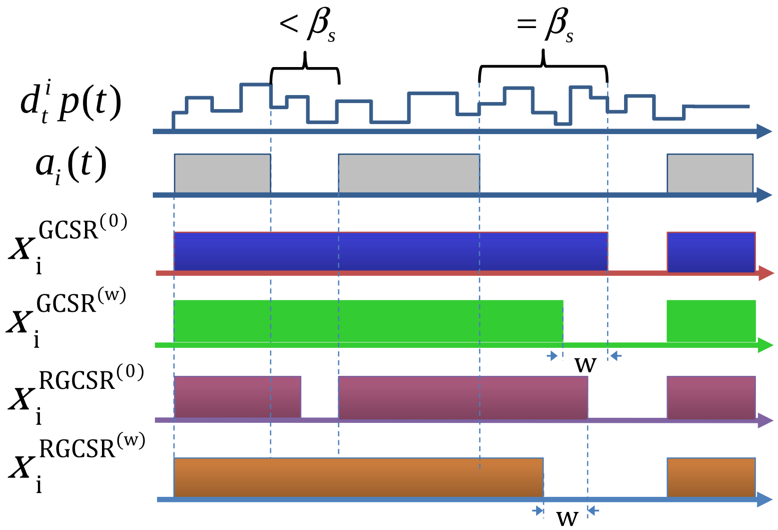


Figure 4.2: An example of $a_i(t)$ and corresponding solutions obtained by $\mathbf{GCSR_s^{(w)}}$ and $\mathbf{RGCSR_s^{(w)}}$. During the first idle period, the cumulative idling cost does not reach β_s, thus $\mathbf{RGCSR_s^{(0)}}$ and $\mathbf{RGCSR_s^{(w)}}$ keep server idling (*i.e.*, $x_i = 1$). Meanwhile, $\mathbf{RGCSR_s^{(0)}}$ turns off the server when the cumulative idling cost reaches the threshold randomly chosen by it. With looking ahead, $\mathbf{RGCSR_s^{(w)}}$ see the job comes before cumulative idling cost reaches β_s, thus it keeps server idling. During the second idle period, $\mathbf{GCSR_s^{(0)}}$ turns off the server when cumulative idling cost reaches β_s. With looking ahead, $\mathbf{GCSR_s^{(w)}}$ turns off the server w slots earlier than $\mathbf{GCSR_s^{(0)}}$. $\mathbf{RGCSR_s^{(0)}}$ turns off the server when cumulative idling cost reaches the threshold it chooses. $\mathbf{RGCSR_s^{(w)}}$ takes actions w slots earlier than $\mathbf{RGCSR_s^{(0)}}$.

4.3 Randomized Online Algorithm RGCSR

Then we develop a randomized online algorithm $\mathbf{RGCSR}^{(w)}$ with a look-ahead window size of w. The idea behind $\mathbf{GCSR_s^{(w)}}$ is to track the offline optimal in an online fashion. However, it is too conservative in making decisions: when it is sure that the idling cost will reach β_s, it turns off the server. In fact, it may be better to be more aggressive in making decisions: we may turn off x_i before seeing idling cost will reach β_s.

More specifically, $\mathbf{RGCSR_s^{(w)}}$ for problem $\mathbf{CP}_\mathrm{i}$ performs as follows: it accumulates an "idling cost" and when it is less than threshold Λ, which is a random variable to be introduced later, it keeps the server idling; otherwise, it will see whether the job will comes, *i.e.*, $a_i > 0$, before the "idling cost" reaches β_s within the look-ahead window w. If so, it keeps the server idling; else it turns off the server. An example is in Fig. 4.2.

The randomized online algorithm for **CP**, denoted as $\mathbf{RGCSR}^{(w)}$, first employs $\mathbf{RGCSR_s^{(w)}}$ to solve each $\mathbf{CP}_\mathrm{i}$, $1 \leq i \leq M$, in an online fashion to produce output $\boldsymbol{x}_i^{on}$ and then simply outputs $\sum_{i=1}^{N} \boldsymbol{x}_i^{on}$ as the output for the original problem **MP**. Formally, $\mathbf{RGCSR^{(w)}}$ is presented in Algorithm 2. The competitive analysis of $\mathbf{RGCSR}^{(w)}$ is in the following theorem.

Denote Λ as a continuous random variables with PDF $f_\Lambda(\lambda)$ (shown in Fig.):

$$f_\Lambda(\lambda) = \begin{cases} \frac{e^{\lambda/[\beta_s(1-\alpha_s)]}}{\beta_s(1-\alpha_s)(e-1+\alpha_s)} + \frac{\alpha_s}{(e-1+\alpha_s)}\delta(\lambda), & \text{if } 0 \leq \lambda \leq (1-\alpha_s)\beta_s, \\ 0, & \text{otherwise,} \end{cases} \tag{4.2}$$

where $\alpha_s = \min\{1, wd_{\min}P_{\min}/\beta_s\}$ and $\delta(\lambda)$ is Dirac Delta function.

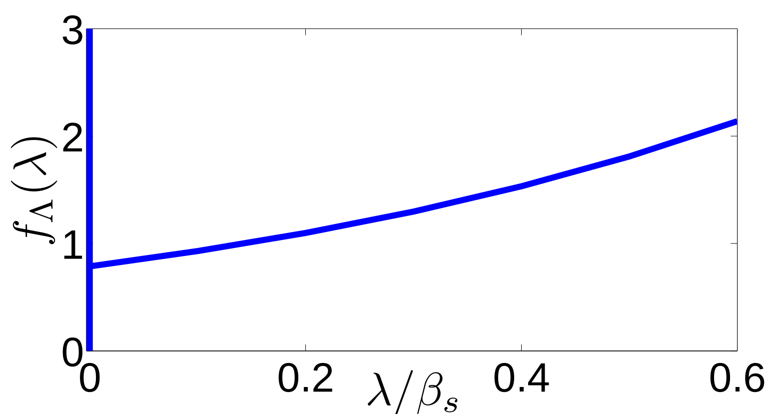


Figure 4.3: PDF $f_\Lambda(\lambda)$. Here $\alpha_s = 0.4$.

Algorithm 2 $\mathbf{RGCSR^{(w)}}$ for problem $\mathbf{CP}_{\mathrm{i}}$

initial condition: $C_i = 0, x_i(0) = 0$, $\forall i \in [1, M]$
at current time $\boldsymbol{t}$, **do**
decompose $(a(\tau))_{\tau=t}^{t+w}$ into $(a_1(\tau), a_2(\tau), \ldots, a_M(\tau))_{\tau=t}^{t+w}$ according to Eqn. (4.1) and obtain $(d_\tau^i)_{\tau=t}^{t+w}$, $\forall i \in [1, M]$
for $i = 1$ to M **do**
  if $a_i(t) > 0$ **then**
    $x_i(t) = 1$, $C_i = 0$, generate Λ according to $f_\Lambda(\lambda)$
  else if $C_i < \Lambda$ **then**
    $x_i(t) = x_i(t-1)$ and $C_i = C_i + p(t)d_t^i x_i(t)$
  else
    set $\tau' \leftarrow \min\{t' \in [t, t+w] \mid C_i + \sum_{\tau=t}^{t'} p(\tau)d_\tau^i \geq \beta_s$ or $a_i(t') > 0\}$
    if $a_i(\tau') > 0$ **then**
      $x_i(t) = x_i(t-1)$ and $C_i = C_i + p(t)d_t^i x_i(t)$
    else
      $x_i(t) = 0$ and $C_i = 0$
    end if
  end if
end for
$x(t) = \sum_{i=1}^{M} x_i(t)$

Theorem 5. $\mathbf{RGCSR}^{(w)}$ *for problem* $\mathbf{CP}$ *has a competitive ratio of* $\frac{e}{e-1+\alpha_s}$*, where* $\alpha_s = \min\{1, w d_{\min} P_{\min}/\beta_s\}$*. Further, no randomized online algorithm with a look-ahead window* w *can achieve a smaller competitive ratio.*

Proof. Refer to Appendix B.3. □

A consequence of Theorem 5 is that when the look-ahead window size w reaches a break-even interval $\Delta_s = \beta_s/(d_{\min} P_{\min})$, $\mathbf{RGCSR}^{(w)}$ has a competitive ratio of 1 as with $\mathbf{GCSR}^{(w)}$.

In addition, Theorems 4 and 5 imply that the worst-case performances of our online algorithms are determined by how much we can see into the future. Algorithms perform better in the worst-case with larger look-ahead window. In practice, we have similar observations: the empirical performances (to be shown in Sec. 6.4) of our algorithms quickly improve to the offline optimal as look-ahead window size w increases. This indicates the value of short-term prediction of inputs to reduce the total cost of the data center.

□ **End of chapter.**

Chapter 5

Online Algorithms for Hybrid Data Centers

Unlike on-grid data centers, hybrid data centers have on-site power generators and therefore have to solve both capacity provisioning (**CP**) and energy provisioning (**EP**) to solve the data center cost minimization (**DCM**) problem. We first briefly present how to solve problem **EP** and then develop our online algorithm **DCMON** for problem **DCM**.

5.1 A useful structure of an offline optimal solution of EP

We first reveal an elegant structure of an offline optimal solution which is very useful in the design of our online algorithm **CHASE**.

5.1.1 Decompose EP into sub-problems EP_is

For the ease of presentation, we denote $e(t) = d_t(x^{on}(t))$. Similar as the decomposition of workload when solving **CP**, we decompose the energy demand $\boldsymbol{e}$ into N sub-demands and define a sub-problem for each generator, then solve energy provisioning *separately* for each sub-problem, where N is the number of on-site generators. Specifically, for $1 \leq i \leq N,\ 1 \leq t \leq T$,

$$e_i(t) \triangleq \min\left\{L, \max\left\{0, e(t) - (i-1)L\right\}\right\}. \tag{5.1}$$

The corresponding sub-problem $\mathbf{EP}_\mathrm{i}$ is in the same form as $\mathbf{EP}$ except that $d_t(\bar{x}(t))$ is replaced by $e_i(t)$ and $y(t)$ is replaced by $y_i(t) \in \{0,1\}$. Using this decomposition, we can solve $\mathbf{EP}$ on input $\boldsymbol{e}$ by simultaneously solving simpler problems $\mathbf{EP}_\mathrm{i}$ on input $\boldsymbol{e}_i$ that only involve a single generator. Theorem 6 shows that the decomposition incurs no optimality loss. Denote $\mathrm{C}_{\mathrm{EP_i}}(\boldsymbol{y_i})$ as the cost of solution $\boldsymbol{y_i}$ for problem $\mathbf{EP}_\mathrm{i}$ and $\mathrm{C}_{\mathrm{EP}}(\boldsymbol{y})$ the cost of solution $\boldsymbol{y}$ for problem $\mathbf{EP}$.

Theorem 6. *Let $\bar{\boldsymbol{y}}_i$ be an optimal solution and $\boldsymbol{y}_i^{on}$ an online solution for $\mathbf{EP_i}$ with energy demand $\boldsymbol{e}_i$, then $\sum_{i=1}^N \bar{\boldsymbol{y}}_i$ is an optimal solution for $\mathbf{EP}$ with energy demand $\boldsymbol{e}$. Furthermore, if $\forall \boldsymbol{e}_i, i$, we have $\mathrm{C}_{\mathrm{EP_i}}(\boldsymbol{y}_i^{on}) \leq \gamma \cdot \mathrm{C}_{\mathrm{EP_i}}(\bar{\boldsymbol{y}}_i)$ for a constant $\gamma \geq 1$, then $\mathrm{C}_{\mathrm{EP}}(\sum_{i=1}^N \boldsymbol{y}_i^{on}) \leq \gamma \cdot \mathrm{C}_{\mathrm{EP}}(\sum_{i=1}^N \bar{\boldsymbol{y}}_i)$, $\forall \boldsymbol{e}$.*

Proof. Refer to Appendix C.1. □

5.1.2 Solve each sub-problem EP_i

Based on Theorem 6, it remains to design algorithms for each $\mathbf{EP}_\mathrm{i}$. Define

$$r_i(t) = \psi\left(0, p(t), e_i(t)\right) - \psi\left(1, p(t), e_i(t)\right). \tag{5.2}$$

$r_i(t)$ can be interpreted as the one-slot cost difference between not using and using an on-site generator. Intuitively, if $r_i(t) > 0$ (resp. $r_i(t) < 0$), it will be desirable to turn on (resp. off) the generator. However, due to the startup cost, we should not turn on and off the generator too frequently. Instead, we should evaluate whether the *cumulative* gain or loss in the future can offset the startup cost. This intuition motivates us to define the following cumulative cost difference $R_i(t)$. We set initial values as $R_i(0) = -\beta_g$ and define $R_i(t)$ inductively:

$$R_i(t) \triangleq \min\left\{0, \max\left\{-\beta_g, R_i(t-1) + r_i(t)\right\}\right\}. \tag{5.3}$$

Note that $R_i(t)$ is only within the range $[-\beta_g, 0]$. An important feature of $R_i(t)$ useful later in online algorithm design is that it can be computed given the past and current inputs. An illustrating example of $R_i(t)$ is shown in Fig. 5.1.

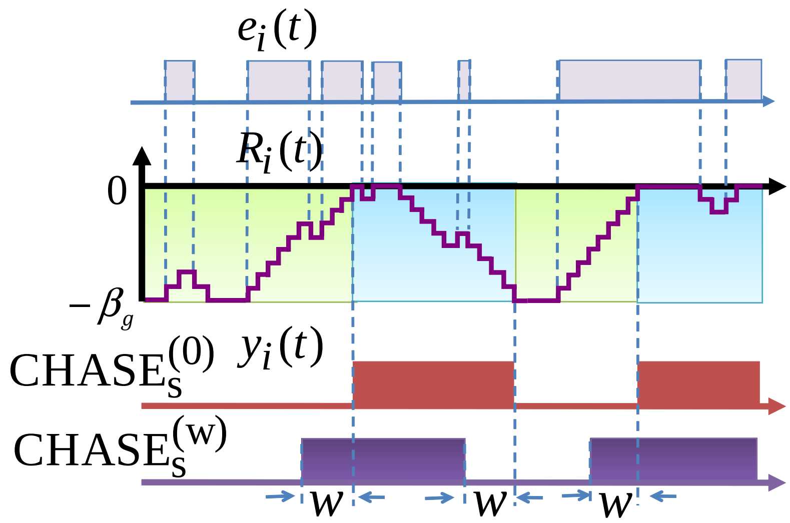


Figure 5.1: An example of $e_i(t)$, $R_i(t)$ and the corresponding solution obtained by $\mathbf{CHASE_s^{(w)}}$ for $\mathbf{EP}_\mathrm{i}$.

Intuitively, when $R_i(t)$ hits its boundary 0, the cost difference between not using and using the generator within a certain period is at least β_g, which can offset the startup cost. Thus, it makes sense to turn on the generator. Similarly, when $R_i(t)$ hits $-\beta_g$, it may be better to turn off the generator and use the grid. The following theorem formalizes this intuition, and shows an optimal solution $\bar{y}_i(t)$ for problem $\mathbf{EP}_\mathrm{i}$ at the time epoch when $R_i(t)$ hits its boundary values $-\beta_g$ or 0.

Theorem 7. *There exists an offline optimal solution for problem* $\mathbf{EP_i}$*, denoted by* $\bar{y}_i(t)$*,* $1 \leq t \leq T$*, so that:*

- *if* $R_i(t) = -\beta_g$*, then* $\bar{y}_i(t) = 0$*;*
- *if* $R_i(t) = 0$*, then* $\bar{y}_i(t) = 1$*.*

Proof. Refer to Appendix C.2. □

5.2 Online algorithm CHASE

The online algorithm $\mathbf{CHASE_s^{(w)}}$ with look-ahead window w exploits the insights revealed in Theorem 7 to solve $\mathbf{EP_i}$. The idea behind $\mathbf{CHASE_s^{(w)}}$ is to track the offline optimal in an online fashion. In particular, at time 0, $R_i(0) = -\beta_g$ and we set $y_i(t) = 0$. We keep tracking the value of $R_i(t)$ at every time slot within the look-ahead window. Once we observe that $R_i(t)$ hits values $-\beta_g$ or 0, we set the $y_i(t)$ to the optimal solution as Theorem 7 reveals; otherwise, keep $y_i(t) = y_i(t-1)$ unchanged. An example of $\mathbf{CHASE_s^{(w)}}$ is shown in Fig. 5.1.

Algorithm 3 $\mathbf{CHASE^{(w)}}$ for problem **EP**

initial condition: $R_i(0) = 0, y_i(0) = 0$, $\forall i \in [1, N]$
at current time $\boldsymbol{t}$, **do**
decompose $(e(\tau))_{\tau=t}^{t+w}$ into $(e_1(\tau), e_2(\tau), \ldots, e_M(\tau))_{\tau=t}^{t+w}$ according to Eqn. (5.1)
for $i = 1$ to N **do**
  obtain $(R_i(\tau))_{\tau=t}^{t+w}$ according to Eqn. (5.3)
  set $\tau' \leftarrow \min\{\tau \in [t, t+w] \mid R_i(\tau) = 0 \text{ or } -\beta_g\}$
  if $\tau' =$ NULL **then**
    $y_i(t) = y_i(t-1)$
  else if $R_i(\tau') = 0$ **then**
    $y_i(t) = 1$
  else
    $y_i(t) = 0$
  end if
end for
$y(t) = \sum_{i=1}^{N} y_i(t)$

The online algorithm for **EP**, denoted as $\mathbf{CHASE^{(w)}}$, first employs $\mathbf{CHASE_s^{(w)}}$ to solve each $\mathbf{EP_i}$ on energy demand $\boldsymbol{e}_i$, $1 \leq i \leq N$, in an online fashion to produce output $\boldsymbol{y}_i^{on}$ and then simply outputs $\sum_{i=1}^{N} \boldsymbol{y}_i^{on}$ as the output for the original problem

EP. More formally, we have Algorithm 3. The competitive analysis of **CHASE**$^{(\mathbf{w})}$ is in Theorem 8.

Theorem 8. **CHASE**$^{(\mathbf{w})}$ *for problem* **EP** *with a look-ahead window* w *has a competitive ratio of*

$$1 + \frac{2\beta_g \left(LP_{\max} - Lc_o - c_m\right)}{\beta_g LP_{\max} + wc_m P_{\max}\left(L - \frac{c_m}{P_{\max} - c_o}\right)}.$$

Proof. Refer to Appendix C.3. □

5.3 Combining GCSR and CHASE

Our online algorithm **DCMON**$^{(\mathbf{w})}$ (Algorithm 4, also shown in Fig. 5.2) for problem **DCM** first uses **GCSR**$^{(\mathbf{w})}$ from Sec. 4.2 to solve problem **CP** and then uses **CHASE**$^{(\mathbf{w})}$ in Sec. 5.2 to solve problem **EP**.

Algorithm 4 **DCMON**$^{(\mathbf{w})}$ for problem **DCM**

1: run **GCSR**$^{(\mathbf{w})}$ to solve **CP** that takes $\left(a(\tau), p(\tau), h(\tau), g_\tau\right)_{\tau=t}^{t+w}$ as inputs and produces the number of active servers $\left(x^{on}(\tau)\right)_{\tau=t}^{t+[w-\Delta_s]^+}$ at each time t.

2: run **CHASE**$^{(\mathbf{w})}$ to solve **EP** that takes $\left(d_\tau(x^{on}(\tau)), p(\tau)\right)_{\tau=t}^{t+[w-\Delta_s]^+}$ as inputs and produces the number of active generators $y^{on}(t)$ at each time t.

As shown in Algorithm 4, an important observation is that the available look-ahead window size for **GCSR**$^{(\mathbf{w})}$ to solve **CP** is w, *i.e.*, knows $p(\tau)$, $a(\tau)$, $h(\tau)$ and g_τ, $1 \leq \tau \leq t + w$, at time t; however, the available look-ahead window size for **CHASE**$^{(\mathbf{w})}$ to solve **EP** is only $[w - \Delta_s]^+$, *i.e.*, knows $p(\tau)$ and $e(\tau) = d_\tau(x^{on}(\tau))$, $1 \leq \tau \leq t + [w - \Delta_s]^+$, at time t ($\Delta_s = \beta_s/(d_{\min}P_{\min})$ is the break-even interval defined in Sec. 4.2).

This is because at time t, **CHASE**$^{(\mathbf{w})}$ knows grid prices $p(\tau)$, workload $a(\tau)$, renewable energy $h(\tau)$ and the function $g_\tau, 1 \leq \tau \leq t + w$. However, not all the energy demands $(e(\tau))_{\tau=1}^{t+w}$ are known by **CHASE**$^{(\mathbf{w})}$. Because we derive the

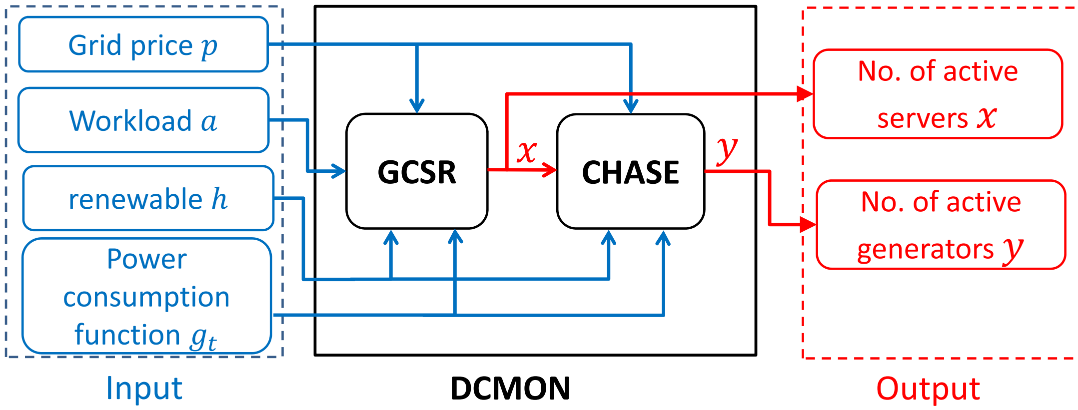


Figure 5.2: The flow chart of **DCMON**.

server state $\boldsymbol{x^{on}}$ by solving problem **CP** using our online algorithm $\mathbf{GCSR^{(w)}}$ using $p(\tau)$, $a(\tau)$, $h(\tau)$, g_τ, $1 \le \tau \le t+w$. A key observation is that at time t it is not possible to compute $\boldsymbol{x^{on}}$ for the full look-ahead window of $t+w$, since $x^{on}(t+1),\ldots,x^{on}(t+w)$ may depend on inputs $p(\tau)$, $a(\tau)$, $h(\tau)$, $g_\tau, \tau > t+w$ that our algorithm does not yet know. Fortunately, for $w \ge \Delta_s$ we can determine all $x^{on}(\tau), 1 \le \tau \le t+[w-\Delta_s]^+$ given inputs within the full look-ahead window. That is, while we knows the grid prices p, the workload a, renewable energy h and the function g_t for the full look-ahead window w, the server state x^{on} is known only for a smaller window of $[w-\Delta_s]^+$. Thus, the energy demand $e(\tau) = d_\tau(x^{on}(\tau)) = [g_\tau(x^{on}(\tau), a(\tau)) - h(\tau)]^+$, $1 \le \tau \le t+[w-\Delta_s]^+$ is available for $\mathbf{CHASE^{(w)}}$ at time t.

The competitive ratio of $\mathbf{DCMON^{(w)}}$ is shown in Theorem 9. It is the product of competitive ratios for $\mathbf{GCSR^{(w)}}$ and $\mathbf{CHASE^{([w-\Delta_s]^+)}}$ from Theorems 4 and 8, respectively, and the optimality loss ratio $LP_{\max}/\left(Lc_o+c_m\right)$ due to the offline-decomposition stated in Sec. 3.2.

Theorem 9. $\mathbf{DCMON^{(w)}}$ *for problem **DCM** has a competitive ratio of*

$$\frac{P_{\max}\left(2-\alpha_s\right)}{c_o+c_m/L}\left[1+\frac{2\left(LP_{\max}-Lc_o-c_m\right)}{LP_{\max}+\alpha_g P_{\max}\left(L-\frac{c_m}{P_{\max}-c_o}\right)}\right].$$

The ratio is also upper-bounded by

$$\frac{P_{\max}(2-\alpha_s)}{c_o + c_m/L}\left[1 + 2\frac{P_{\max} - c_o}{P_{\max}} \cdot \frac{1}{1+\alpha_g}\right],$$

where $\alpha_s = \min(1, w/\Delta_s) \in [0,1]$ *and* $\alpha_g \triangleq \frac{c_m}{\beta_g}[w-\Delta_s]^+ \in [0, +\infty)$ *are "normalized" look-ahead window sizes.*

Proof. Refer to Appendix D. □

As the look-ahead window size w increases, the competitive ratio in Theorem 9 decreases to $LP_{\max}/(Lc_o + c_m)$ (c.f. Fig. 5.3), the inherent optimality loss introduced by our decomposition approach stated in Sec. 3.2. However, the real trace based empirical performance [1] of $\mathbf{DCMON^{(w)}}$ is quite close to the offline optimal, *i.e.*, ratio close to 1 (c.f. Fig. 5.3).

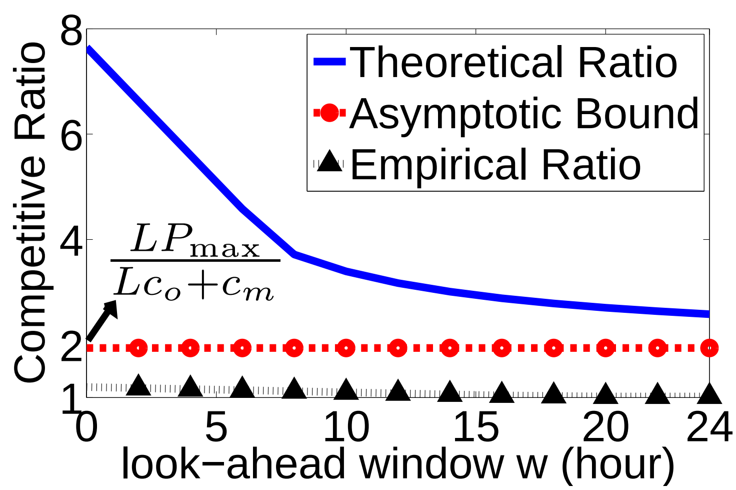


Figure 5.3: Theoretical and empirical ratios of algorithm $\mathbf{DCMON^{(w)}}$ vs. look-ahead window size w.

This is because in practice the inputs are usually not the worst case inputs to our online algorithm and thus the empirical ratio is much smaller than the theoretical ratio. In practice, our online algorithm aligns quite well with the offline optimal. As we can see from Fig. 5.4, for generator #1, $R_1(t)$ (cumulative cost difference

[1] The parameter settings are in Chapter 6.

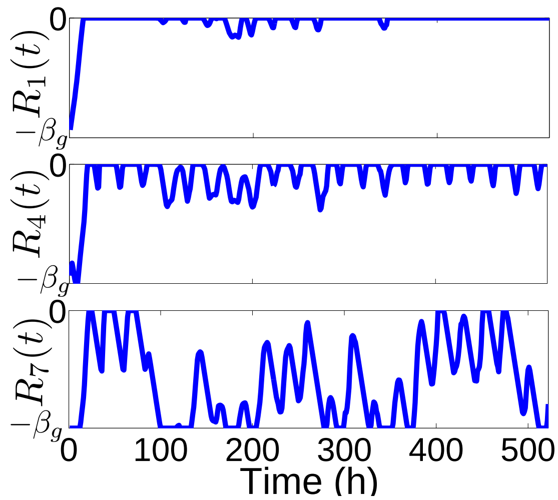


Figure 5.4: Cumulative cost difference functions (Eqn. 5.3) of three generators in real-world trace based simulations.

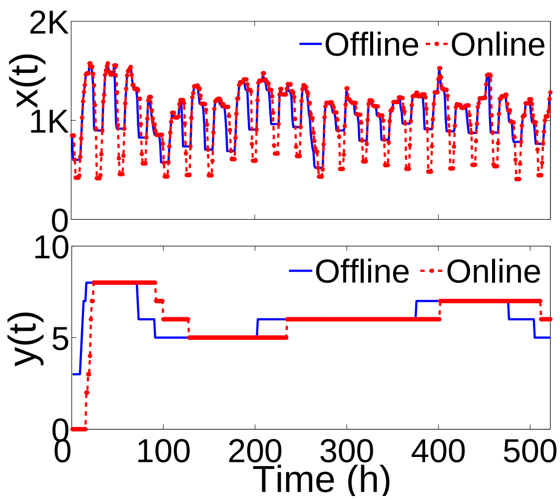


Figure 5.5: Comparison of $x(t), y(t)$ obtained by **DCMON$^{(\mathbf{w})}$** and the offline optimal.

function) remains 0 most of the time and does not decrease to $-\beta_g$. Thus, both our online and the offline optimal keep the generator on most of the time. Similar observations can be found for generator #4. For generator #7, although $R_7(t)$ goes to 0 some time and decreases to $-\beta_g$ some time, it does not bounce forth and back between the two boundaries very often. Thus, our online can still align with the offline optimal well. For servers, we have similar observation and we skip the details here.

In addition, from Fig. 5.5, we see that our online solution (*i.e.*, the number of active servers $x(t)$ and the number of active on-site generators $y(t)$) always track the offline optimal. Thus, the former lags behind the latter. However, most of the time, our online performs exactly the same as the offline optimal. Consequently, the cost achieved by our online is quite close to that achieved by the offline optimal.

□ **End of chapter.**

Chapter 6

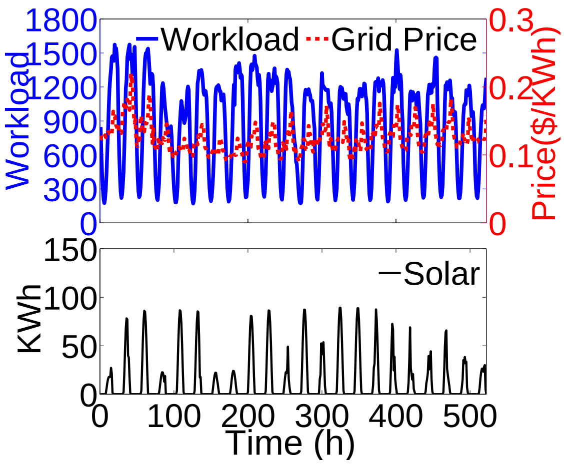
Empirical Evaluations

In this chapter, we evaluate the performance of our algorithms by simulations based on real-world traces with the aim of (i) corroborating the empirical performance of our online algorithms under various realistic settings and the impact of having look-ahead information, (ii) understanding the benefit of opportunistically procuring energy from on-site generators, the grid and the renewable sources, as compared to the current practice of using the grid and the renewable sources, (iii) studying how much on-site generation is needed for substantial cost benefits, (iv) studying the impact of renewable penetration.

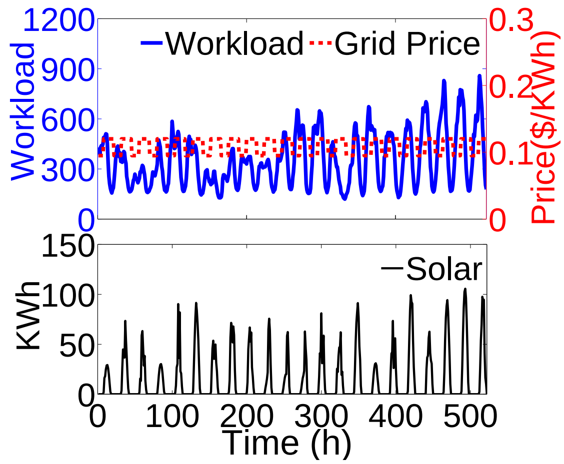
6.1 Parameters and Settings

Workload trace: We use the workload traces from the Akamai network [1, 32] that is the currently the world's largest content delivery network. Specifically, the traces record the hourly average load served by each deployed server in New York and San Jose over 22 days from Dec. 21, 2008 to Jan. 11, 2009. The New York trace represents 2.5K servers that served about 1.4×10^{10} requests and 1.7×10^{13} bytes of content to end-users during our measurement period. The San Jose trace represents 1.5K servers that served about 5.5×10^{9} requests and 8×10^{12} bytes of content. We show the workload in Fig. 6.1, in which we normalize the load by the server capacity.

Grid price: We use traces of hourly grid power prices in New York [3] and San Jose [4] for the same time period, so that it can be matched up with the workload traces (c.f. Fig. 6.1). Both workload and grid price traces show strong diurnal properties: in the daytime, the workload and the grid price are relatively high; at



(a) New York



(b) San Jose

Figure 6.1: Real-world workload from Akamai and the corresponding grid power prices and renewable energy within the same time period.

night, on the contrary, both are low. This indicates the feasibility of reducing the data center cost by using the energy from the on-site generators during the daytime and using the grid at night.

Renewable trace: We use traces of hourly solar radiation (wh/m^2) in New York central park and San Jose international airport observation stations within the same time period (Dec. 21, 2008 to Jan. 11, 2009) from National Solar Radiation Data Base [2]. By assuming a solar panel array with area $1000m^2$ and a conversion rate of 20% [5], we obtain the renewable energy $h(t)$ (c.f. Fig. 6.1). Roughly speaking, for the New York trace, the renewable supplies about 4% of the overall energy demand. For the San Jose trace, the number is about 10%.

Server model: Similar to a typical setting in [34], we use the standard linear server power consumption model. According to [34], each server consumes 0.25KWh power per hour at full capacity and has a power proportional factor (PPF=$(c_{peak} - c_{idle})/c_{peak}$) of 0.6, which gives us $c_{idle} = 0.1KW$, $c_{peak} = 0.25KW$. In addition, we assume the server switching cost equals the energy cost of running a server for 3 hours. If we assume an average grid price as the price of energy, we get about

$\beta_s = \$0.08$.

Cooling and power conditioning system model: We consider a water chiller cooling system. According to [7], during this 22-day winter period the average high and low temperatures of New York are $41°F$ and $29°F$, respectively. Those of San Jose are $58°F$ and $41°F$, respectively. We take the high temperature as the daytime temperature and the low temperature as the nighttime temperature. Thus, according to [35], the power consumed by water chiller cooling systems of the New York and San Jose data centers are about

$$f^t_{c,NY}(b) = \begin{cases} (0.041b^2 + 0.144b + 0.047)b_{\max}, & \text{at daytime}, \\ (0.03b^2 + 0.136b + 0.042)b_{\max}, & \text{at nighttime}, \end{cases}$$

and

$$f^t_{c,SJ}(b) = \begin{cases} (0.06b^2 + 0.16b + 0.054)b_{\max}, & \text{at daytime}, \\ (0.041b^2 + 0.144b + 0.047)b_{\max}, & \text{at nighttime}, \end{cases}$$

where $b_{\max}$ is the maximum server power consumption and b is the server power consumption normalized by $b_{\max}$. The maximum server power consumption of the New York and San Jose data centers are $b^{NY}_{\max} = 2500 \times 0.25 = 625KW$ and $b^{SJ}_{\max} = 1500 \times 0.25 = 375KW$. Besides, the power consumed by the power conditioning system, including PDUs and UPSs, is $f_p(b) = (0.012b^2 + 0.046b + 0.056)b_{\max}$ [35].

Generator model: We adopt generators with specifications the same as the one in [6]. The maximum output of the generator is 60KW, *i.e.*, $L = 60KW$. The incremental cost c_o is set to be \$0.08/KWh, which is calculated according to the gas price [3] and the generator efficiency [6]. Similar to [38], we set the sunk cost (maintenance cost) c_m to be \$1.2. We set the startup cost β_g equivalent to the amortized capital cost, which gives $\beta_g = \$24$. Besides, we assume the number of generators $N = 10$, which is enough to satisfy all the energy demand for the trace and model we use.

Cost benchmark: Current data centers usually do not use dynamic capacity provisioning and on-site generators. Thus, we use the cost incurred by static capacity provisioning with only grid power as the benchmark. This static provisioning runs a fixed number of servers, without dynamically turning on/off the servers, that minimizes the costs incurred based on full knowledge of the entire workload. Using such a benchmark gives a conservative evaluation of the cost reduction achieved by our algorithms.

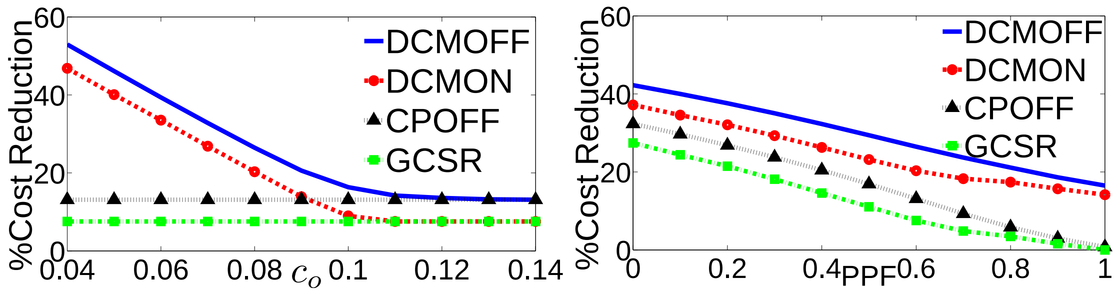
Comparisons of Algorithms: We consider two scenarios: the on-grid and hybrid scenarios. In each scenario, we compare three algorithms: the offline optimal algorithm, our online algorithm and the Receding Horizon Control (RHC) algorithm (also summarized in Tab. 6.1). RHC is a heuristic algorithm commonly used in the control literature [25]. It works by solving, at each time t, the cost minimization problem over window $[t, t+w]$ given the initial state $x(t-1), y(t-1)$, and then using the first step of the solution, discarding the rest. We note that because at each step RHC does not consider any adversarial future dynamics beyond the time-window w, there is no guarantee that RHC is competitive.

	On-gird	Hybrid
Offline Optimal	**CPOFF**	**DCMOFF**
Our Online	**GCSR** & **RGCSR**	**DCMON**
RHC	**CPRHC**	**DCMRHC**

Table 6.1: Compared Algorithms.

6.2 Impact of Model Parameters on Cost Reduction

We study the cost reduction provided by our offline and online algorithms for both on-grid and hybrid data centers using the New York trace unless specified otherwise. We assume no look-ahead information is available when running the online algorithms. We compute the cost reduction (in percentage) as compared to the cost benchmark which we described earlier. When all parameters take their default



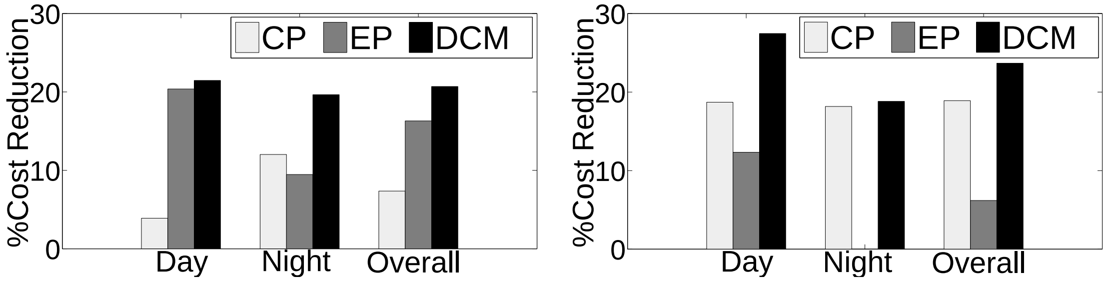
(a) Cost Reduction vs. $c_o(\$/KWh)$ (b) Cost Reduction vs. PPF (PPF=$(c_{peak} - c_{idle})/c_{peak}$)

Figure 6.2: Variation of cost reduction with model parameters.

values, our offline (resp. online) algorithms provide up to 13.1% (resp., 7.5%) cost reduction for on-grid and 26.5% (resp., 20.3%) cost reduction for hybrid data centers (c.f. Fig. 6.2). Note that the online algorithms provide cost reduction that are 6% smaller than offline algorithms on account of their lack of knowledge of future inputs. Further, note that cost reduction of a hybrid data center is larger than that of a on-grid data center, since hybrid data center has the ability to generate energy on-site to avoid higher grid prices. Nevertheless, the extent of cost reduction in all cases is high providing strong evidence for the need to perform energy and server capacity optimizations.

Data centers may deploy different types of servers and generators with different model parameters. It is then important to understand the impact on cost reduction due to these parameters. We first study the impact of varying c_o (c.f. Fig. 6.2). For a hybrid data center, as c_o increases the cost of on-site generation increases making it less effective for cost reduction (c.f Fig. 6.2a). For the same reason, the cost reduction of a hybrid data center tends to that of the on-grid data center with increasing c_o as on-site generation becomes less economical.

We then study the impact of power proportional factor (PPF). More specifically, we fix $c_{peak} = 0.25KW$, and vary PPF from 0 to 1 (c.f. Fig. 6.2b). As PPF increases, the server idle power decreases, thus dynamic server provisioning has lesser impact on the cost reduction. This explains why **CP** achieves no cost reduction



(a) New York (b) San Jose

Figure 6.3: Relative values of CP, EP, and DCM.

when PPF=1. Since **DCM** involves **CP** problem, its performance degrades with increasing PPF as well.

6.3 The Relative Value of Energy versus Capacity Provisioning

In this subsection, we use both New York and San Jose traces. For a hybrid data center, we ask which optimization provides a larger cost reduction: energy provisioning (**EP**) or server capacity provisioning (**CP**) in comparison with the joint optimization of doing both (**DCM**). The cost reductions of different optimization are shown in Fig. 6.3.

For the New York scenario in Fig. 6.3a, overall, we see that **EP**, **CP**, and **DCM** provide cost reductions of 16%, 7.5%, and 20.3%, respectively. However, note that during the day doing **EP** alone provides almost as much cost reduction as the joint optimization **DCM**. The reason is that during the high traffic hours in the day, solving **EP** to avoid higher grid prices provides a larger benefit than optimizing the energy consumption by server shutdown. The opposite is true during the night where **CP** is more critical than **EP**, since minimizing the energy consumption by shutting down idle servers yields more benefit.

For the San Jose scenario in Fig. 6.3b, overall, **EP**, **CP**, and **DCM** provide cost reductions of 6.1%, 19%, and 23.7%, respectively. Compared to the New York scenario, the reason why **EP** achieves so little cost reduction is that the grid power is cheaper and thus on-site generation is not that economical. Meanwhile, **CP** performs closer to **DCM**, which is because the workload curve is highly skew (shown in Fig. 6.1b) and dynamic server provisioning saves a lot of server idling cost as well as cooling and power conditioning cost.

In a nutshell, **EP** favors high grid power price while workload with less regular pattern makes **CP** more competitive.

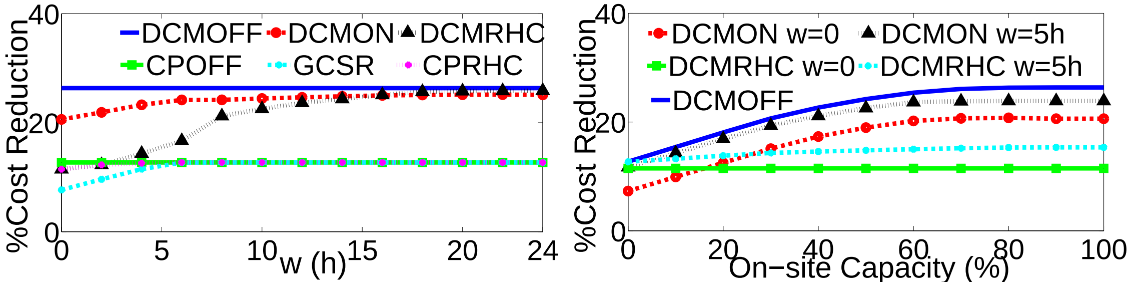
6.4 Benefit of Look-ahead

We evaluate the cost reduction benefit of increasing the look-ahead window. From Fig. 6.4a, we observe that the performance of our online algorithms **DCMON** and **GCSR** are already very good when there is no or little look-ahead information (*e.g.,* $w = 0$, 2, and 4). In contrast, **DCMRHC** performs poorly when the look-ahead window is small. When w is large, (*e.g.,* $w \geq 8$), both **DCMON** and **DCMRHC** performs close to the offline optimal **DCMOFF**.

An interesting observation is that it is more important to design intelligent online algorithms when no or little look-ahead information is available. When there is abundant look-ahead information, both **DCMON** and **DCMRHC** achieve good performance and it is less critical to carry out sophisticated algorithm design.

6.5 How Much On-site Power Production is E-nough

Thus far, in our experiments, we assumed that a hybrid data center had the ability to supply all its energy from on-site power generators ($N = 10$). However, an important question is how much investment should a data center operator make in installing on-site generator capacity to obtain largest cost reduction.



(a) Cost Reduction vs. look-ahead window size w

(b) Cost Reduction vs. percentage of on-site power production capacity

Figure 6.4: Variation of cost reduction with look-ahead and on-site capacity.

More specifically, we vary the number of on-site generators N from 0 to 10 and show how the algorithms perform. Interestingly, in Fig. 6.4b, our results show that provisioning on-site generators to produce 80% of the peak power demand of the data center is sufficient to obtain all of the cost reduction benefits. Further, with just 60% on-site power generation capacity we can achieve 95% of the maximum cost reduction. The intuitive reason is that most of time the power demand of the data center is significantly lower than their peaks.

6.6 Impact of Renewable Penetration

We evaluate the impact of renewable penetration on the data center cost reduction. More specifically, we vary the renewable penetration by scaling up/down the solar trace and show how our algorithms perform. Note that renewable penetration $X\%$ means the total renewable energy is $X\%$ of the total energy consumption.

From Fig. 6.5, we observe that the performances of **DCMOFF** and **DCMON** degrade as renewable penetration increases. Intuitively, this is because with more renewable energy, the net energy demand needed to be satisfied by the grid or on-site generators is lower. Consequently, energy provisioning is less important. Meanwhile, the performances of **CPOFF** and **GCSR** are hardly influenced by the renewable penetration. The reason is that the benchmark algorithm (static

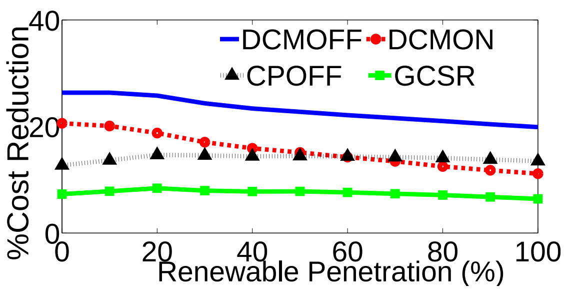


Figure 6.5: Impact of renewable penetration.

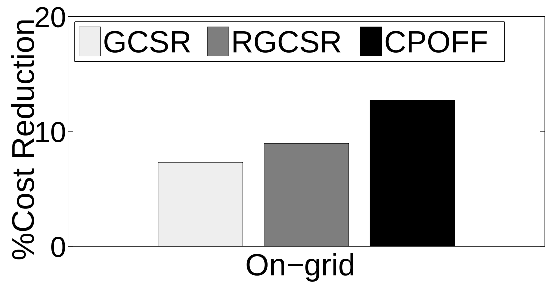


Figure 6.6: Randomized v.s. Deterministic Algorithms.

provisioning) and capacity provisioning (**CPOFF** and **GCSR**) benefit roughly the same from increasing renewable penetration, and thus the cost reductions of capacity provisioning against the benchmark remain almost unchanged.

6.7 Randomized Algorithms vs. Deterministic Algorithms

We evaluate performances of our randomized online algorithm **RGCSR** for the on-grid scenario. The cost reductions compared to the benchmark are shown in Fig. 6.6.

The randomized (resp. deterministic) algorithm achieves 9% (resp., 7.3%) cost reduction for on-grid data centers. The randomized algorithm performs a little better than the deterministic algorithm in both scenarios. This is because randomized algorithm is more aggressive in decision making as discussed in Sec. 4.3. It aligns with the offline optimal better and thus achieve a larger cost reduction.

□ **End of chapter.**

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This work focuses on the cost minimization of data centers achieved by jointly optimizing *both* the supply of energy from on-site power generators, the grid and the renewable sources, and the demand for energy from its deployed servers as well as power conditioning and cooling systems. We show that such an integrated approach is not only possible in next-generation data centers but also desirable for achieving significant cost reductions. Our offline optimal algorithm and our online algorithms with provably good competitive ratios provide key ideas on how to coordinate energy procurement and production with the energy consumption.

Our empirical work answers several of the important questions relevant to data center operators focusing on minimizing their operating costs. We show that an on-grid and hybrid data center can achieve a cost reduction between 7.5% to 13.1% and 20.3% to 26.5%, respectively, by employing our joint optimization framework. We also show that on-site power generation can provide an additional cost reduction of about 13%, and that most of the additional benefit is obtained by a partial on-site generation capacity of 60% of the peak power requirement of the data center.

7.2 Discussion

Previously, we assume homogeneous servers and homogeneous generators. However, in reality, a data center may consist of multiple types of servers as well as generators. Thus, it is of great interest to study the heterogeneous setting. In this section, we discuss how to extend our algorithm **DCMON** to be able to solve data center cost minimization problem in heterogeneous scenario.

Notation	**Definition**
Ts	Number of types of servers
N_s^i	Number of type-i servers, $i \in \{1, \ldots, Ts\}$
β_s^i,L_s^i	Parameters of a type-i server, L_s^i is the service capacity
$x^i(t)$	Number of active type-i servers
$a^i(t)$	Workload distributed to type-i servers
$f_s^i\left(x^i(t), a^i(t)\right)$	Type-i servers' power consumption as a function of $x^i(t)$ and $a^i(t)$
Tg	Number of types of generators
N_g^j	Number of type-j generators, $j \in \{1, \ldots, Tg\}$
β_g^j,L_g^j ,c_m^j,c_o^j	Parameters of a type-j generator
$y^j(t)$,$u^j(t)$	Variables associated with type-j generators

Table 7.1: Notation for the heterogeneous scenario.

7.2.1 The Heterogeneous Scenario

Based on Tab. 2.1, we add superscripts i, j to denote parameters and variables of different types of servers and generators (c.f. Tab. 7.1).

According to Chapter 2 and above notations, the total power consumption g_t is:

$$\begin{aligned} & \sum_{i=1}^{Ts} f_s^i\left(x^i(t), a^i(t)\right) + f_p\left(\sum_{i=1}^{Ts} f_s^i\left(x^i(t), a^i(t)\right)\right) + f_c^t\left(\sum_{i=1}^{Ts} f_s^i\left(x^i(t), a^i(t)\right)\right) \\ \triangleq\ & g_t\left(x^1(t), a^1(t), \ldots, x^{Ts}(t), a^{Ts}(t)\right) \end{aligned}$$

Our objective is to minimize the total cost of the data center in entire horizon

$[1, T]$, which is given by

$$\text{Cost}(x, y, u, v) \triangleq \sum_{t=1}^{T} \left\{ v(t)p(t) + \sum_{i=1}^{Ts} \beta_s^i [x^i(t) - x^i(t-1)]^+ \right. \\ \left. + \sum_{j=1}^{Tg} \left(c_o^j u^j(t) + c_m^j y^j(t) + \beta_g^j [y^j(t) - y^j(t-1)]^+ \right) \right\}.$$

Then the data center cost minimization problem in heterogeneous scenario is

DCM-HET :

$$\min_{x,y,s,u,v} \quad \text{Cost}(x, y, u, v)$$

$$\text{s.t.} \quad \sum_{j=1}^{Tg} u^j(t) + v(t) + h(t) \geq g_t \left(x^1(t), a^1(t), \ldots, x^{Ts}(t), a^{Ts}(t) \right), \tag{7.1}$$

$$\sum_{i=1}^{Ts} a^i(t) = a(t), \tag{7.2}$$

$$u^j(t) \leq L_g^j y^j(t),$$

$$L_s^i x(t) \geq a^i(t),$$

$$y^j(t) \leq N_g^j, \; x^i(t) \leq N_s^i,$$

$$x^i(0) = 0, \; y^j(0) = 0,$$

$$\text{var} \quad x^i(t), y^j(t) \in \mathbb{N}^0, u^j(t), v(t), a^i(t) \in \mathbb{R}_0^+, \; i \in [1, Ts], j \in [1, Tg], t \in [1, T].$$

Note that this problem is much more challenging than **DCM** to solve. Because it involves multiple types of servers and generators. Different types of generators are correlated by constraint (7.1). Correlation between different types of servers are imposed by constraint (7.2). This kind of correlation makes it difficult for us to design online algorithms with provable good performance.

7.2.2 A Heuristic Extension of DCMON

We extend the online algorithm **DCMON** designed for hybrid data center with homogeneous servers and generators to the heterogeneous scenario. To begin with, we define *efficiency index*:

$$E_s^i \triangleq f_s^i(N_s^i, N_s^i \cdot L_s^i)/\left(N_s^i \cdot L_s^i\right), \ \forall i \in \{1, \ldots, Ts\}$$

and

$$E_g^j \triangleq c_o^j + c_m^j/L_g^j, \ \forall j \in \{1, \ldots, Tg\},$$

where E_s^i measures the average power consumption of type-i servers to process one unit workload and E_g^j measures the average cost of type-j generators to output one unit energy. Lower power consumption or cost means higher efficiency. Without loss of generality, we assume $E_s^1 \leq E_s^2 \leq \cdots \leq E_s^{Ts}$ and $E_g^1 \leq E_g^2 \leq \cdots \leq E_g^{Tg}$, *i.e.*, the one with smaller index has smaller *efficiency index*.

Intuitively, the servers with small E_s^i and the generators with small E_g^j should be used with high priority, because of their low power consumption/cost. Based on this idea, our online algorithm solves problem **DCM-HET** as follows: at each time, the servers with small E_s^i and generators with small E_g^j are used first. In another word, workload is distributed to servers with small E_s^i first and energy demand is distributed to generators with small E_g^i first. Unless all the ones with smaller *efficiency index* are used, the ones with larger *efficiency index* will not be used.

More formally, the workload distributed to type-i servers is $\forall t \in \{1, \ldots, T\}, \ i \in \{1, \ldots, T_s\}$,

$$a^i(t) \triangleq \min\left\{N_s^i L_s^i, \max\left\{0, a(t) - \sum_{k=1}^{i-1} N_s^i L_s^i\right\}\right\}.$$

Denote $e(t)$ as the total net energy demand at t, then the energy demand distributed to type-j generators is $\forall t \in \{1, \ldots, T\}, \ j \in \{1, \ldots, T_g\}$,

$$e^j(t) \triangleq \min\left\{N_g^j L_g^j, \max\left\{0, e(t) - \sum_{k=1}^{j-1} N_s^j L_s^j\right\}\right\}.$$

Our algorithm $\mathbf{DCMON_{het}^{(w)}}$ for solving problem **DCM-HET** with a look-ahead window of $w \geq 0$ works as follows.

1. Run $\mathbf{GCSR^{(w)}}$ from Sec. 4.2 to solve a series of **CP**s, each of which is with N_s^i type-i servers, workload $\boldsymbol{a^i}$ and $d_t(x^i(t)) = [g_t\left(N_s^1, N_s^1 L_s^1, \ldots, N_s^{i-1}, N_s^{i-1} L_s^{i-1},\right.$

$x^i(t), a^i(t), 0, 0, \ldots) - h(t)]^+,\ 1 \le i \le Ts$. We obtain number of active servers of each type $\boldsymbol{x^1}, \boldsymbol{x^2}, \ldots, \boldsymbol{x^{Ts}}$. And the energy demand is determined by $e(t) = [g_t\left(x^1(t), s^1(t), \ldots, x^{Ts}(t), s^{Ts}(t)\right) - h(t)]^+$.

2. Run algorithm $\mathbf{CHASE^{(w)}}$ from Sec. 5.2 to solve a series of **EP**s, each of which is with N_g^j type-j generators and energy demand $\boldsymbol{e^j},\ 1 \le j \le Tg$. We obtain number of active generators of each type $\boldsymbol{y^1}, \boldsymbol{y^2}, \ldots, \boldsymbol{y^{Tg}}$. Besides, the output from each type of generators $u^j(t)$ and power drawn from the grid $v(t)$ is determined by

$$u^j(t) = \begin{cases} 0, & \text{if } p(t) \le c_o^j, \\ \min\left(L_g^j y^j(t), e^j(t)\right), & \text{otherwise,} \end{cases}$$

and

$$v(t) = e(t) - \sum_{j=1}^{Tg} u^j(t).$$

7.2.3 Empirical Performances

Servers	Type	N_s	$\beta_s(\$)$	L_s	$c_{idle}(KW)$	$c_{peak}(KW)$
	1	700	0.08	1	0.1	0.25
	2	700	0.05	0.5	0.06	0.14
	3	700	0.15	1.5	0.16	0.45
Generators	Type	N_g	$\beta_g(\$)$	$L_g(KW)$	$c_m(\$)$	$c_o(\$/KWh)$
	1	3	24	60	1.2	0.08
	2	3	46	100	1.8	0.07
	3	4	14	80	2	0.09

Note that L_s is normalized by the service capacity of the Type-1 server.

Table 7.2: Parameters of servers and generators.

We use the same parameter settings as Sec. 6.1 except parameters of servers and generators. Here, we assume the data center has 3 types of servers and 3 types of generators, whose parameters are shown in Tab. 7.2.

We first evaluate the cost reduction benefit of increasing the look-ahead window obtained by $\mathbf{DCMON_{het}}$ and $\mathbf{DCMRHC_{het}}$ and compare it with $\mathbf{DCMOFF_{het}}$. From Fig. 7.1a, we observe that while the performance of $\mathbf{DCMON_{het}}$ is already good when there is no look-ahead information, it quickly improves to the offline optimal when a small amount of look-ahead is available. In contrast, $\mathbf{DCMRHC_{het}}$ performs poorly when the look-ahead window is small as with the homogeneous case in Sec. 6.4. Note that while $\mathbf{DCMON_{het}}$ is a heuristic extension of **DCMON** to the heterogeneous scenario, it performs very close to the offline optimal for realistic inputs. The reason may be that $\mathbf{DCMON_{het}}$ always uses servers and generators with small *efficiency index* (defined in Sec. 7.2.2) first. Remark that *efficiency index* for server measures the average power consumption of a type of servers to process one unit workload and *efficiency index* for generators measures the average cost of a type of generators to output one unit energy. This "smallest first" strategy of $\mathbf{DCMON_{het}}$ may not be optimal but is very cost efficient in most cases.

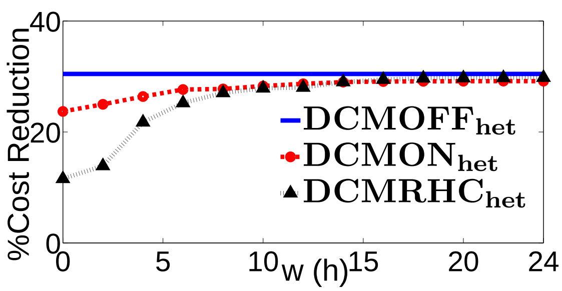
We then evaluate performance of $\mathbf{DCMON_{het}}$ against $\mathbf{DCMOFF_{het}}$ under different heterogeneity settings. More specifically, based on Tab. 7.2, we fix the parameters of type-1 server and type-1 generator and set the parameters of other types of servers and generators in Tab. 7.3. We vary values of γ_s and γ_g from 1 to 4 and show the corresponding ratio of $\mathbf{DCMON_{het}}$ against $\mathbf{DCMOFF_{het}}$. Note that the larger γ_s and γ_g are, the larger the differences between different servers. So are the generators.

From Fig. 7.1b, we observe that the ratio between $\mathbf{DCMON_{het}}$ and $\mathbf{DCMOFF_{het}}$ increases as the heterogeneity parameters γ_s and γ_g increase. Even so, the performance of $\mathbf{DCMON_{het}}$ is always within about 1.2 times the offline optimal $\mathbf{DCMOFF_{het}}$, which demonstrates the effectiveness of our online algorithm $\mathbf{DCMON_{het}}$.

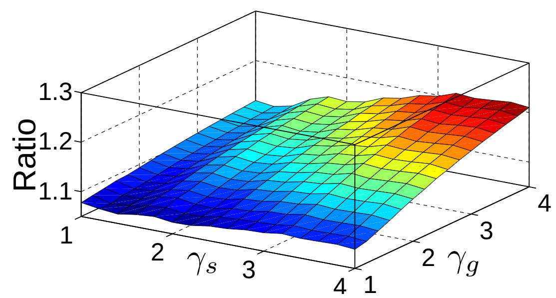
Servers	Type	N_s	β_s	L_s	c_{idle}	c_{peak}
	1	700	0.08	1	0.1	0.25
	2	700	$0.08\gamma_s$	1	$0.1/\gamma_s$	$0.25/\gamma_s$
	3	700	$0.08/\gamma_s$	1	$0.1\gamma_s$	$0.25\gamma_s$
Generators	Type	N_g	β_g	L_g	c_m	c_o
	1	3	24	60	1.2	0.08
	2	3	$24\gamma_g$	60	$1.2/\gamma_g$	$0.08/\gamma_g$
	3	4	$24/\gamma_g$	60	$1.2\gamma_g$	$0.08\gamma_g$

$\gamma_s, \gamma_g \geq 1$ measures the heterogeneity of servers and generators, respectively.

Table 7.3: Parameters of servers and generators.



(a) Cost Reduction vs. look-ahead window w in the heterogeneous scenario



(b) Ratio between algorithms **DCMON$_{het}$** and **DCMOFF$_{het}$** vs. heterogeneity parameters γ_s and γ_g

Figure 7.1: Performance in the heterogeneous scenario.

7.3 Future Work

There are several promising directions for future research. First, it is interesting to study the competitive ratio of the algorithm for heterogeneous scenarios proposed in Sec. 7.2.2. Second, studying how energy storage devices can be used to further reduce the data center operating cost is of great interest. Third, it is desirable to generalize our analysis to take into account deferrable workloads.

☐ **End of chapter.**

Appendix A

Proof of Theorem 2

First, we prove that the factor loss in optimality is at most $LP_{\max}/\left(Lc_o+c_m\right)$.

Then, we prove that the factor loss is tight.

Let $(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})$ be the solution obtained by solving **CP** and **EP** separately in sequence and $(\boldsymbol{x}^*,\boldsymbol{y}^*)$ be the solution obtained by solving the joint-optimization **DCM**. Denote $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x},\boldsymbol{y})$ to be cost of **DCM** of solution $(\boldsymbol{x},\boldsymbol{y})$ and $\mathrm{C}_{\mathrm{CP}}(\boldsymbol{x})$ to be cost of **CP** of solution $\boldsymbol{x}$.

It is straightforward that

$$\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})\leq\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}},\mathbf{0}).\tag{A.1}$$

Because $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x},0)=\mathrm{C}_{\mathrm{CP}}(\boldsymbol{x})$, we have

$$\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}},\mathbf{0})=\mathrm{C}_{\mathrm{CP}}(\bar{\boldsymbol{x}})\leq\mathrm{C}_{\mathrm{CP}}(\boldsymbol{x}^*)=\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*,\mathbf{0}).\tag{A.2}$$

By Eqns. (A.1) and (A.2), we obtain

$$\frac{\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})}{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*,\boldsymbol{y}^*)}\leq\frac{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*,\mathbf{0})}{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*,\boldsymbol{y}^*)}.\tag{A.3}$$

Then, according to the following lemma, we get

$$\rho=\frac{\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})}{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*,\boldsymbol{y}^*)}\leq\frac{LP_{\max}}{Lc_o+c_m}.$$

Lemma 1. $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*,\mathbf{0})/\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*,\boldsymbol{y}^*)\leq LP_{\max}/\left(Lc_o+c_m\right).$

Proof. By plugging solutions $(\boldsymbol{x}^*,\mathbf{0})$ and $(\boldsymbol{x}^*,\boldsymbol{y}^*)$ into **DCM** separately, we have

$$\begin{aligned}\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*,\mathbf{0}) \;=\; & \sum_{t=1}^{T}\{p(t)d_t\left(x^*(t)\right)\\ & +\beta_s[x^*(t)-x^*(t-1)]^+\}\end{aligned}\tag{A.4}$$

and

$$\begin{aligned}
\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*) &= \sum_{t=1}^{T} \{\psi\left(y^*(t), p(t), d_t\left(x^*(t)\right)\right) \\
&\quad +\beta_s[x^*(t) - x^*(t-1)]^+ \\
&\quad +\beta_g[y^*(t) - y^*(t-1)]^+\big\} \\
&\geq \sum_{t=1}^{T} \{\psi\left(y^*(t), p(t), d_t\left(x^*(t)\right)\right) \\
&\quad +\beta_s[x^*(t) - x^*(t-1)]^+\big\}. \qquad \text{(A.5)}
\end{aligned}$$

By Eqns. (A.4), (A.5) and (2.9), we obtain

$$\begin{aligned}
&\frac{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{0})}{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*)} \\
\leq\ & \frac{\sum_{t=1}^{T} p(t) d_t\left(x^*(t)\right)}{\sum_{t=1}^{T} \psi\left(y^*(t), p(t), d_t\left(x^*(t)\right)\right)} \\
\leq\ & \max_{t\in\{1,..,T\}} \frac{p(t) d_t\left(x^*(t)\right)}{\psi\left(y^*(t), p(t), d_t\left(x^*(t)\right)\right)} \\
\leq\ & \begin{cases} 1, & \text{if } p(t) \leq c_o, \\ \frac{P_{\max} d_t(x^*(t))}{c_o d_t(x^*(t)) + c_m \lceil d_t(x^*(t))/L \rceil}, & \text{otherwise} \end{cases} \\
\leq\ & \frac{P_{\max} d_t\left(x^*(t)\right)}{c_o d_t\left(x^*(t)\right) + c_m d_t\left(x^*(t)\right)/L} \\
=\ & \frac{P_{\max}}{c_o + c_m/L}.
\end{aligned}$$

□

Next, we prove that the factor loss is tight.

Lemma 2. *There exist an input such that* $\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})/\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*) = LP_{\max}/\left(Lc_o + c_m\right)$.

Proof. Consider the following input:

$$g_t(x, a) = e_m x,\ h(t) = 0,\ p(t) = P_{\max},\ \forall t,$$

and

$$a(t) = \begin{cases} \frac{L}{e_m}, & \text{if } t = 1 + k(1 + \frac{\beta_s}{e_m P_{\max}}),\ k \in \mathbb{N}^0, \\ 0, & \text{otherwise,} \end{cases}$$

where $e_m > 0$ is a constant such that L/e_m is an integer.

Then for the above input and setting, it is not difficult to see that an optimal solution to problem **CP** is given by

$$\bar{x}(t) = \begin{cases} \frac{L}{e_m}, & \text{if } t = 1 + k(1 + \frac{\beta_s}{e_m P_{\max}}),\ k \in \mathbb{N}^0, \\ 0, & \text{otherwise.} \end{cases}$$

Besides, we can also see that the following $\boldsymbol{x}^*$ is an optimal solution to problem **CP** and also problem **DCM** whatever y^* is.

$$x^*(t) = \frac{L}{e_m},\ \forall t.$$

Intuitively, this is because with on-site generation in problem **DCM**, unit power cost may be reduced from $P_{\max}$ to c_o, consequently, servers tends to be idling longer when there is no workload.

Then, we consider the following parameter setting:

$$L(P_{\max} - c_o) - c_m < \beta_g,$$

$$L(P_{\max} - c_o) - c_m - \frac{\beta_s}{e_m P_{\max}} c_m < 0,$$

and

$$L(P_{\max} - c_o) - c_m - \frac{\beta_s}{e_m P_{\max}} c_m + \frac{\beta_s L}{e_m P_{\max}}(P_{\max} - c_o) > 0.$$

Since $\bar{\boldsymbol{x}}$ and $\boldsymbol{x}^*$ have been determined by us, we can apply Theorem 10 to obtain the corresponding $\bar{\boldsymbol{y}}$ and $\boldsymbol{y}^*$. According to the definition of $R_i(t)$ (Eqn. (5.3)) and the above parameter setting, given $\bar{\boldsymbol{x}}$ and $\boldsymbol{a}$, the corresponding $R_i(t)$ never reaches 0. However, given $\boldsymbol{x}^*$ and $\boldsymbol{a}$, the corresponding $R_i(t)$ will soon reach 0 and never fall back to $-\beta_g$. So we have

$$\bar{y}(t) = 0 \text{ and } y^*(t) = 1, \forall t.$$

See Fig. A.1 as an example. By plugging the above $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ and $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ into

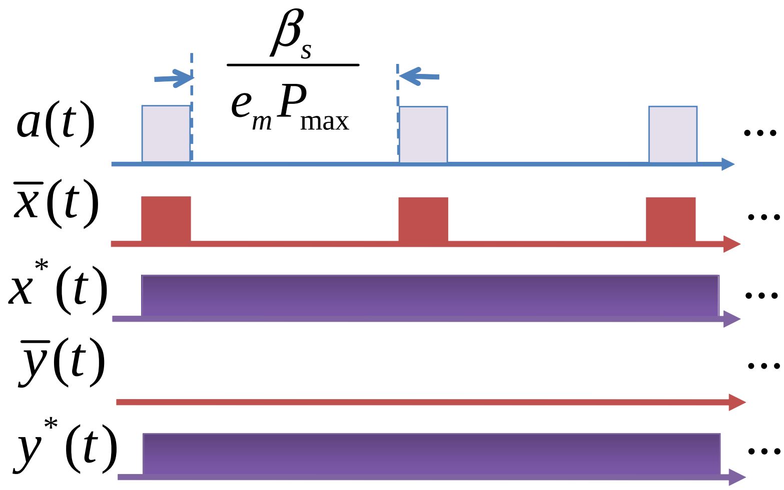


Figure A.1: An example of $a(t)$, $\bar{x}(t)$, $x^*(t)$, $\bar{y}(t)$ and $y^*(t)$. $\bar{x}(t)$, $\bar{y}(t)$ are obtained by solving **CP** and **EP** sequentially. $x^*(t)$, $y^*(t)$ are obtained by solving **DCM** directly.

DCM, we have

$$\begin{aligned}\frac{\mathrm{C_{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})}{\mathrm{C_{DCM}}(\boldsymbol{x}^*, \boldsymbol{y}^*)} &= \frac{LP_{\max} + \beta_s L/e_m}{Lc_o + c_m + (Lc_o + c_m)\beta_s/(e_m P_{\max})} \\ &= \frac{LP_{\max}\left[1 + \beta_s/(e_m P_{\max})\right]}{(Lc_o + c_m)\left[1 + \beta_s/(e_m P_{\max})\right]} \\ &= \frac{LP_{\max}}{Lc_o + c_m}.\end{aligned}$$

□

Theorem 2 follows from Eqn. (A.3), lemmas 1 and 2.

□ **End of chapter.**

Appendix B

Proofs of Theorems 3, 4 and 5

B.1 Proof of Theorem 3

First, we show that the combined solution $\sum_{i=1}^{M} \bar{\boldsymbol{x}}_i$ is optimal to **CP**.

Denote $\mathrm{C_{CP}}(\boldsymbol{x})$ to be cost of **CP** of solution $\boldsymbol{x}$. Suppose that $\tilde{\boldsymbol{x}}$ is an optimal solution for **CP**. We will show that we can construct a new feasible solution $\sum_{i=1}^{M} \hat{\boldsymbol{x}}_i$ for **CP**, and a new feasible solution $\hat{\boldsymbol{x}}_i$ for each $\mathbf{CP}_\mathrm{i}$, such that

$$\mathrm{C_{CP}}(\tilde{\boldsymbol{x}}) = \mathrm{C_{CP}}(\sum_{i=1}^{M} \hat{\boldsymbol{x}}_i) = \sum_{i=1}^{M} \mathrm{C_{CP_i}}(\hat{\boldsymbol{x}}_i) + \sum_{t=1}^{T} p(t) d_t(0). \tag{B.1}$$

$\bar{\boldsymbol{x}}_i$ is an optimal solution for each $\mathbf{CP}_\mathrm{i}$. Hence, $\mathrm{C_{CP_i}}(\hat{\boldsymbol{x}}_i) \geq \mathrm{C_{CP_i}}(\bar{\boldsymbol{x}}_i)$ for each i. Thus,

$$\begin{aligned} \mathrm{C_{CP}}(\tilde{\boldsymbol{x}}) &= \sum_{i=1}^{M} \mathrm{C_{CP_i}}(\hat{\boldsymbol{x}}_i) + \sum_{t=1}^{T} p(t) d_t(0) \\ &\geq \sum_{i=1}^{M} \mathrm{C_{CP_i}}(\bar{\boldsymbol{x}}_i) + \sum_{t=1}^{T} p(t) d_t(0). \end{aligned} \tag{B.2}$$

Besides, we also can prove that

$$\sum_{i=1}^{M} \mathrm{C_{CP_i}}(\bar{\boldsymbol{x}}_i) + \sum_{t=1}^{T} p(t) d_t(0) \geq \mathrm{C_{CP}}(\sum_{i=1}^{M} \bar{\boldsymbol{x}}_i). \tag{B.3}$$

Hence, $\mathrm{C_{CP}}(\tilde{\boldsymbol{x}}) = \mathrm{C_{CP}}(\sum_{i=1}^{M} \bar{\boldsymbol{x}}_i)$, *i.e.*, $\sum_{i=1}^{M} \bar{\boldsymbol{x}}_i$ is an optimal solution for **CP**.

Then, we show $\mathrm{C_{CP}}(\sum_{i=1}^{M} \boldsymbol{x}_i^{on}) \leq \gamma \cdot \mathrm{C_{CP}}(\sum_{i=1}^{M} \bar{\boldsymbol{x}}_i)$.

Because $\mathrm{C_{CP_i}}(\boldsymbol{x}_i^{on}) \leq \gamma \cdot \mathrm{C_{CP_i}}(\bar{\boldsymbol{x}}_i)$ and $\bar{\boldsymbol{x}}_i$ is optimal for $\mathbf{CP}_\mathrm{i}$, we have $\gamma \geq 1$. According to Eqn. (B.2), we obtain

$$\gamma \cdot \mathrm{C_{CP}}(\tilde{\boldsymbol{x}}) \geq \sum_{i=1}^{M} \mathrm{C_{CP_i}}(\boldsymbol{x}_i^{on}) + \sum_{t=1}^{T} p(t) d_t(0).$$

Besides, we also can prove that

$$\sum_{i=1}^{M} \mathrm{C}_{\mathrm{CP_i}}(\boldsymbol{x}_i^{on}) + \sum_{t=1}^{T} p(t)d_t(0) \;\; \geq \;\; \mathrm{C}_{\mathrm{CP}}(\sum_{i=1}^{M} \boldsymbol{x}_i^{on}). \tag{B.4}$$

Hence, $\mathrm{C}_{\mathrm{CP}}(\sum_{i=1}^{M} \boldsymbol{x}_i^{on}) \leq \gamma \cdot \mathrm{C}_{\mathrm{CP}}(\sum_{i=1}^{M} \bar{\boldsymbol{x}}_i)$.

It remains to prove Eqns. (B.1), (B.3) and (B.4), which we show in Lemmas 3 and 4.

Lemma 3. $\mathrm{C}_{\mathrm{CP}}(\tilde{\boldsymbol{x}}) = \mathrm{C}_{\mathrm{CP}}(\sum_{i=1}^{M} \hat{\boldsymbol{x}}_i) = \sum_{i=1}^{M} \mathrm{C}_{\mathrm{CP_i}}(\hat{\boldsymbol{x}}_i) + \sum_{t=1}^{T} p(t)d_t(0)$.

Proof. Define $\hat{\boldsymbol{x}}_i$ based on $\tilde{\boldsymbol{x}}$ by:

$$\widehat{x}_i(t) = \begin{cases} 1, & \text{if } i \leq \tilde{x}(t) \\ 0, & \text{otherwise.} \end{cases}$$

It is straightforward to see that

$$\tilde{x}(t) = \sum_{i=1}^{M} \hat{x}_i(t)$$

and $\hat{\boldsymbol{x}}_i$ is a feasible solution for $\mathbf{CP}_{\mathrm{i}}$, i.e., $\hat{\boldsymbol{x}}_i \geq \boldsymbol{a}_i$.

So we have $\mathrm{C}_{\mathrm{CP}}(\tilde{\boldsymbol{x}}) = \mathrm{C}_{\mathrm{CP}}(\sum_{i=1}^{M} \hat{\boldsymbol{x}}_i)$.

Note that $\widehat{x}_1(t) \geq ... \geq \widehat{x}_M(t)$ is a decreasing sequence. Because $\widehat{x}_i(t) \in \{0,1\}, \; \forall i,t$, we obtain

$$\begin{aligned} & \sum_{i=1}^{M} [\hat{x}_i(t) - \hat{x}_i(t-1)]^+ \\ = \; & \begin{cases} 0, & \text{if } \sum_{i=1}^{M} \hat{x}_i(t) \leq \sum_{i=1}^{M} \hat{x}_i(t-1) \\ \sum_{i=1}^{M} \hat{x}_i(t) - \sum_{i=1}^{M} \hat{x}_i(t-1), & \text{otherwise} \end{cases} \\ = \; & \Big[\sum_{i=1}^{M} \hat{x}_i(t) - \sum_{i=1}^{M} \hat{x}_i(t-1) \Big]^+, \end{aligned} \tag{B.5}$$

and

$$\begin{aligned}\sum_{i=1}^{M} d_t^i \cdot \hat{x}_i(t) + d_t(0) &= \sum_{i=1}^{\tilde{x}(t)} d_t^i \cdot 1 + d_t(0) \\ &= \sum_{i=1}^{\tilde{x}(t)} [d_t(i) - d_t(i-1)] + d_t(0) \\ &= d_t(\tilde{x}(t)) - d_t(0) + d_t(0) \\ &= d_t(\tilde{x}(t)) = d_t(\sum_{i=1}^{M} \hat{x}_i(t)). \qquad \text{(B.6)}\end{aligned}$$

By Eqns. (B.5) and (B.6),

$$\mathrm{C_{CP}}(\sum_{i=1}^{M} \hat{\boldsymbol{x}}_i) = \sum_{i=1}^{M} \mathrm{C_{CP_i}}(\hat{\boldsymbol{x}}_i) + \sum_{t=1}^{T} p(t) d_t(0).$$

This completes the proof of this lemma. □

Lemma 4. $\sum_{i=1}^{M} \mathrm{C_{CP_i}}(\boldsymbol{x}_i) + \sum_{t=1}^{T} p(t) d_t(0) \geq \mathrm{C_{CP}}(\sum_{i=1}^{M} \boldsymbol{x}_i)$*, where* $\boldsymbol{x}_i$ *is any feasible solution for problem* $\boldsymbol{CP}_{\mathrm{i}}$*.*

Proof. First, it is straightforward that

$$\sum_{i=1}^{M} [x_i(t) - x_i(t-1)]^+ \geq \Big[\sum_{i=1}^{M} x_i(t) - \sum_{i=1}^{M} x_i(t-1)\Big]^+. \qquad \text{(B.7)}$$

Denote $x(t) = \sum_{i=1}^{M} x_i(t)$. Then, $\forall t$,

$$\begin{aligned}\sum_{i=1}^{M} d_t^i \cdot x_i(t) + d_t(0) &\geq \sum_{i=1}^{x(t)} d_t^i + d_t(0) \\ &= d_t(x(t)) - d_t(0) + d_t(0) \\ &= d_t(x(t)) = d_t(\sum_{i=1}^{M} x_i(t)), \qquad \text{(B.8)}\end{aligned}$$

where the first inequality comes from $x_i(t) \in \{0, 1\}$ and $d_t^1 \leq d_t^2 \leq \cdots \leq d_t^M$. This is because $d_t^i = d_t(i) - d_t(i-1)$ and $d_t(x)$ is convex in x.

This lemma follows from Eqns. (B.7) and (B.8). □

B.2 Proof of Theorem 4

First, we will characterize an offline optimal algorithm for $\mathbf{CP}_{\mathrm{i}}$.

Then, based on the optimal algorithm, we prove the competitive ratio of our future-aware online algorithm $\mathbf{GCSR}_{\mathbf{s}}^{(\mathbf{w})}$.

Finally, we prove the lower bound of competitive ratio of any deterministic online algorithm.

In $\mathbf{CP}_{\mathrm{i}}$, the workload input $\boldsymbol{a_i}$ takes value in $[0, 1]$ and exactly one server is required to serve each $\boldsymbol{a_i}$. When $a_i(t) > 0$, we must keep $x_i(t) = 1$ to satisfy the feasibility condition. The problem is what we should do if the server is already active but there is no workload, i.e., $a_i(t) = 0$.

To illustrate the problem better, we define *idling interval* I_1 as follows: $I_1 \triangleq [t_1, t_2]$, such that (i) $a_i(t_1 - 1) > 0$; (ii) $a_i(t_2 + 1) > 0$; (iii) $\forall \tau \in [t_1, t_2]$, $a_i(\tau) = 0$. Similarly, define the *working interval* I_2*:* $I_2 \triangleq [t_1, t_2]$, such that (i) $a_i(t_1 - 1) = 0$; (ii) $a_i(t_2 + 1) = 0$; (iii) $\forall \tau \in [t_1, t_2]$, $a_i(\tau) > 0$. Define the *starting interval* Is*:* $I_s \triangleq [0, t_2]$, such that (i) $a_i(t_2 + 1) > 0$; (ii) $\forall \tau \in [0, t_2]$, $a_i(\tau) = 0$. Define the *ending interval* I_e*:* $I_e \triangleq [t_1, T + 1]$, such that (i) $a_i(t_1 - 1) > 0$; (ii) $\forall \tau \in [t_1, T + 1]$, $a_i(\tau) = 0$.

Based on the above definitions, we have the following offline optimal algorithm $\mathbf{CPOFF}_{\mathbf{s}}$ for problem $\mathbf{CP}_{\mathrm{i}}$.

Lemma 5. $\mathbf{CPOFF}_{\mathbf{s}}$ *is an offline optimal algorithm to problem* $\boldsymbol{CP}_{\mathrm{i}}$*.*

Proof. It is easy to see that it is optimal to set $x_i = 0$ during I_s and I_e and set $x_i = 1$ during each I_2.

During an I_1, an offline optimal solution must set either $x_i(\tau) = 0$ or $x_i(\tau) = 1, \forall \tau \in I_1$; otherwise, it will incur unnecessary switching cost and can not be optimal. The cost of setting $x_i = 1$ during an I_1 is $\sum_{t \in I_1} d_t^i p(t)$. The cost of setting $x_i = 0$ during I_1 is β_s, because we must pay a turn-on cost β_s after this I_1. Thus the above algorithm $\mathbf{CPOFF}_{\mathbf{s}}$ is an offline optimal algorithm to $\mathbf{CP}_{\mathrm{i}}$. $\square$

Algorithm 5 An offline optimal Algorithm $\mathbf{CPOFF_s}$ for $\mathbf{CP}_{\mathrm{i}}$

According to $\boldsymbol{a_i}$, find I_s, I_e and all the I_1 and I_2.
During I_s and I_e , set $x_i = 0$.
During each I_2, set $x_i = 1$.
During each I_1,
if $\sum_{t \in I_1} p(t) d_t^i \geq \beta_s$ **then**
    set $x_i(\tau) = 0, \forall \tau \in I_1$.
else
    set $x_i(\tau) = 1, \forall \tau \in I_1$.
end if

Lemma 6. $\mathbf{GCSR_s^{(w)}}$ *is* $(2 - a_s)$*-competitive for problem* $\boldsymbol{CP}_{\mathrm{i}}$*, where* $\alpha_s \triangleq \min(1, w d_{\min} P_{\min}/\beta_s) \in [0,1]$ *and* $d_{\min} \triangleq \min_t \{d_t(1) - d_t(0)\} \geq 0$*. Hence, according to Theorem 3,* $\mathbf{GCSR^{(w)}}$ *achieves the same competitive ratio for* $\boldsymbol{CP}$*.*

Proof. We compare our online algorithm $\mathbf{GCSR_s^{(w)}}$ and the offline optimal algorithm $\mathbf{CPOFF_s}$ described above for problem $\mathbf{CP}_{\mathrm{i}}$ and prove the competitive ratio. Let $\boldsymbol{x_i^{on}}$ and $\bar{\boldsymbol{x}}_i$ be the solutions obtained by $\mathbf{GCSR_s^{(w)}}$ and $\mathbf{CPOFF_s}$ for problem $\mathbf{CP}_{\mathrm{i}}$, respectively.

Since $d_t(x(t))$ is increasing and convex in $x(t)$, we have

$$\begin{aligned} d_t^i &= d_t(i) - d_t(i-1) \\ &\geq d_t(i-1) - d_t(i-2) \\ &\vdots \\ &\geq d_t(1) - d_t(0) \\ &\geq \min_t \{d_t(1) - d_t(0)\} = d_{\min} \geq 0. \end{aligned} \tag{B.9}$$

Let $\bar{\boldsymbol{x}}_i$ and $\boldsymbol{x}_i^{on}$ be the solutions obtained by $\mathbf{CPOFF_s}$ and $\mathbf{GCSR_s^{(w)}}$, respectively. for problem $\mathbf{CP_i}$. It is easy to see that during I_s and I_2, $\mathbf{GCSR_s^{(w)}}$ and $\mathbf{CPOFF_s}$ have the same actions. Since the adversary can choose the T to be large enough, we can omit the cost incurred during I_e when doing competitive analysis.

Thus, we only need to consider the cost incurred by the $\mathbf{GCSR_s^{(w)}}$ and $\mathbf{CPOFF_s}$ during each I_1. Notice that at the beginning of an I_2, both algorithm may incur switching cost. However, there must be an I_1 before an I_2. So this switching cost will be taken into account when we analyze the cost incurred during I_1. More formally, for a certain I_1,denoted as $[t_1, t_2]$,

$$\begin{aligned} & Cost_{I_1}(\boldsymbol{x_i}) \\ = & \sum_{t=t_1}^{t_2} p(t) d_t^i \left(x_i(t) - \lceil a_i(t) \rceil\right) + \beta_s \sum_{t=t_1}^{t_2+1} \left[x_i(t) - x_i(t-1)\right]^+ \\ = & \sum_{t=t_1}^{t_2} p(t) d_t^i x_i(t) + \beta_s \sum_{t=t_1}^{t_2+1} \left[x_i(t) - x_i(t-1)\right]^+ . \end{aligned} \tag{B.10}$$

$\mathbf{GCSR_s^{(w)}}$ performs as follows: it accumulates an "idling cost" and when it reaches β_s, it turns off the server; otherwise, it keeps the server idle. Specifically, at time t, if there exists $\tau \in [t, t+w]$ such that the idling cost till τ is at least β_s, it turns off the server; otherwise, it keeps it idle. We distinguish two cases:

Case 1: $w \geq \beta_s/(d_{\min} P_{\min})$. In this case, $\mathbf{GCSR_s^{(w)}}$ performs the same as $\mathbf{CPOFF_s}$. Because

If $\sum_{t \in I_1} d_t^i p(t) \geq \beta_s$, $\mathbf{CPOFF_s}$ turns off the server at the beginning of the I_1, i.e., at t_1. Since $w \geq \beta_s/(d_{\min} P_{\min})$ and $d_t^i \geq d_{\min}$ according to Eqn. (B.9), at t_1 $\mathbf{GCSR_s^{(w)}}$ can find a $\tau \in [t_1, t_1 + w]$ such that the idling cost till τ is at least β_s, as a consequence of which it also turns off the server at the beginning of the I_1. Both algorithms turn on the server at the beginning of the following I_2. Thus, we obtain

$$Cost_{I_1}(\boldsymbol{x_i^{on}}) = Cost_{I_1}(\bar{\boldsymbol{x}}_i) = \beta_s. \tag{B.11}$$

If $\sum_{t \in I_1} d_t^i p(t) < \beta_s$,$\mathbf{CPOFF_s}$ keeps the server idling during the whole I_1. $\mathbf{GCSR_s^{(w)}}$ finds that the accumulate idling cost till the end of the I_1 will not reach β_s, so it also keeps the server idling during the whole I_1. Thus, we have

$$Cost_{I_1}(\boldsymbol{x_i^{on}}) = Cost_{I_1}(\bar{\boldsymbol{x}}_i) = \sum_{t \in I_1} d_t^i p(t).$$

Case 2: $w < \beta_s/(d_{\min}P_{\min})$. In this case, to beat $\mathbf{GCSR_s^{(w)}}$, the adversary will choose $p(t)$, $a_i(t)$ and d_t^i so that $\mathbf{GCSR_s^{(w)}}$ will keep the server idling for some time and then turn it off, but $\mathbf{CPOFF_s}$ will turn off the server at the beginning of the I_1. Suppose $\mathbf{GCSR_s^{(w)}}$ keeps the server idling for δ slots given no workload within the look-ahead window and then turn it off. Then according to Algorithm 1, we must have $\sum_{\delta+w} d_t^i p(t) < \beta_s$ and $\sum_{\delta+w+1} d_t^i p(t) \geq \beta_s$. In this case, $Cost_{I_1}(\bar{\boldsymbol{x}}_i) = \beta_s$ and

$$\begin{aligned} Cost_{I_1}(\boldsymbol{x_i^{on}}) &= \sum_{\delta} d_t^i p(t) + \beta_s \\ &= \sum_{\delta+w} d_t^i p(t) - \sum_{w} d_t^i p(t) + \beta_s \\ &\leq \beta_s - d_{\min}P_{\min}w + \beta_s \\ &= \beta_s(2 - \frac{d_{\min}P_{\min}}{\beta_s}w). \end{aligned}$$

So

$$\begin{aligned} \frac{\mathrm{C_{CP_i}}(\boldsymbol{x_i^{on}})}{\mathrm{C_{CP_i}}(\bar{\boldsymbol{x}}_i)} &\leq \frac{Cost_{I_1}(\boldsymbol{x_i^{on}})}{Cost_{I_1}(\bar{\boldsymbol{x}}_i)} \\ &\leq 2 - \frac{d_{\min}P_{\min}}{\beta_s}w. \end{aligned}$$

Combining the above two cases establishes this lemma.

Furthermore, we have some important observations on $\boldsymbol{x_i^{on}}$ and $\bar{\boldsymbol{x}}_i$, which will be used in later proofs.

$$\sum_{t=1}^{T} \left[x_i^{on}(t) - x_i^{on}(t-1)\right]^+ = \sum_{t=1}^{T} \left[\bar{x}_i(t) - \bar{x}_i(t-1)\right]^+. \tag{B.12}$$

This is because during an I_1 with $\sum_{t\in I_1} d_t^i p(t) \geq \beta_s$, $\boldsymbol{x_i^{on}}$ keeps the server idling for some time and then turn it off. $\bar{\boldsymbol{x}}_i$ turns off the server at the beginning of the I_1. Both $\boldsymbol{x_i^{on}}$ and $\bar{\boldsymbol{x}}_i$ turn on the server at the beginning of the following I_2. During an I_1 with $\sum_{t\in I_1} d_t^i p(t) < \beta_s$, both $\boldsymbol{x_i^{on}}$ and $\bar{\boldsymbol{x}}_i$ keep the server idling till the following I_2. Thus, $\boldsymbol{x_i^{on}}$ and $\bar{\boldsymbol{x}}_i$ incur the same server switching cost. Besides, in both above cases, $x_i^{on}(t)$ is no less than $\bar{x}_i(t)$, we have

$$\boldsymbol{x_i^{on}} \geq \bar{\boldsymbol{x}}_i. \tag{B.13}$$

We also observe that

$$\begin{aligned} & \sum_{t=1}^{T} d_t^i p(t) \left(x_i^{on}(t) - \lceil a_i(t) \rceil\right) \\ \leq \; & \sum_{t=1}^{T} d_t^i p(t) \left(\bar{x}_i(t) - \lceil a_i(t) \rceil\right) + \\ & (1-\alpha_s) \sum_{t=1}^{T} \left[\bar{x}_i(t) - \bar{x}_i(t-1)\right]^+ . \end{aligned} \tag{B.14}$$

By rearranging the terms, we obtain

$$\sum_{t=1}^{T} d_t^i p(t) \left(x_i^{on}(t) - \bar{x}_i(t)\right) \leq (1-\alpha_s) \sum_{t=1}^{T} \left[\bar{x}_i(t) - \bar{x}_i(t-1)\right]^+ . \tag{B.15}$$

Notice that $\sum_{t=1}^{T} d_t^i p(t) \left(x_i(t) - \lceil a_i(t) \rceil\right)$ can be seen as the total server idling cost incurred by solution $\boldsymbol{x}_i$. Since idling only happens in I_1, Eqn. (B.14) follows from the cases discussed above. □

Lemma 7. $(2 - a_s)$ *is the lower bound of competitive ratio of any deterministic online algorithm for problem* $\boldsymbol{CP}_{\mathrm{i}}$ *and also* $\boldsymbol{CP}$*, where* $\alpha_s \triangleq \min\left(1, w d_{\min} P_{\min}/\beta_s\right) \in [0, 1]$.

Proof. First, we show this lemma holds for problem $\mathbf{CP}_{\mathrm{i}}$. We distinguish two cases:

Case 1: $w \geq \beta_s/(d_{\min} P_{\min})$. In this case,$(2 - a_s) = 1$, which is clearly the lower bound of competitive ratio of any online algorithm.

Case 2: $w < \beta_s/(d_{\min} P_{\min})$. Similar as the proof of Lemma 6, we only need to analyze behaviors of online and offline algorithms during an *idle interval* I_1.

Consider the input: $d_t^i = d_{\min}$ and $p(t) = P_{\min}$,$\forall t \in [1, T]$. Under this input, during an I_1, we only need to consider a set of deterministic online algorithms with the following behavior: either keep the server idling for the whole I_1 or keep it idling for some slots and then turn if off until the end of the I_1. The reason is that any deterministic online algorithm not belonging to this set will turn off the server at some time and turn on the server before the end of I_1, and thus there must be an online algorithm incurring less cost by turning off the server at the same time but turning on the server at the end of I_1.

We characterize an algorithm **ALG** belonging to this set by a parameter δ, denoting the time it keeps the server idling for given $a_i \equiv 0$ within the lookahead window. Denote the solutions of algorithms **ALG** and $\mathbf{CPOFF_s}$ for problem $\mathbf{CP}_\mathrm{i}$ to be $\boldsymbol{x_i^{alg}}$ and $\bar{\boldsymbol{x}}_i$, respectively.

If δ is infinite, the competitive ratio is apparently infinite due to the fact that the adversary can construct an I_1 whose duration is infinite. Thus we only consider those algorithms with finite δ. The adversary will construct inputs as follows:

If $\delta + w \geq \beta_s/(d_{\min}P_{\min})$, the adversary will construct an I_1 whose duration is longer than $\delta + w$. In this case, **ALG** will keep server idling for δ slots and then turn if off while $\mathbf{CPOFF_s}$ turns off the server at the beginning of the I_1 (c.f. Fig. B.1a). Then the ratio is

$$\begin{aligned}
\frac{\mathrm{C_{CP_i}}(\boldsymbol{x_i^{alg}})}{\mathrm{C_{CP_i}}(\bar{\boldsymbol{x}}_i)} &= \frac{\sum_\delta d_{\min}P_{\min} + \beta_s + d_{\min}P_{\min}}{\beta_s + d_{\min}P_{\min}} \\
&> 1 + \frac{[\beta_s/(d_{\min}P_{\min}) - w]\, d_{\min}P_{\min}}{\beta_s + d_{\min}P_{\min}} \\
&= 2 - \frac{d_{\min}P_{\min}(w+1)}{\beta_s + d_{\min}P_{\min}}.
\end{aligned}$$

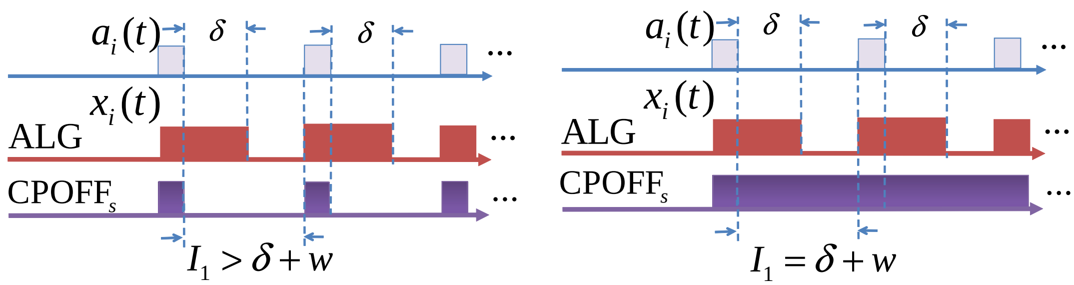
If $\delta + w < \beta_s/(d_{\min}P_{\min})$, the adversary will construct an I_1 whose duration is exactly $\delta + w$. In this case, **ALG** will keep server idling for δ slots and then turn if off while $\mathbf{CPOFF_s}$ keeps the server idling during the whole I_1 (c.f. Fig. B.1b). Then the ratio is

$$\begin{aligned}
\frac{\mathrm{C_{CP_i}}(\boldsymbol{x_i^{alg}})}{\mathrm{C_{CP_i}}(\bar{\boldsymbol{x}}_i)} &= \frac{\sum_\delta d_{\min}P_{\min} + \beta_s + d_{\min}P_{\min}}{d_{\min}P_{\min}(\delta + w) + d_{\min}P_{\min}} \\
&= \frac{d_{\min}P_{\min}(\delta + w + 1) + \beta_s - w d_{\min}P_{\min}}{d_{\min}P_{\min}(\delta + w + 1)} \\
&\geq 1 + \frac{\beta_s - w d_{\min}P_{\min}}{\beta_s + d_{\min}P_{\min}} \\
&= 2 - \frac{d_{\min}P_{\min}(w+1)}{\beta_s + d_{\min}P_{\min}}.
\end{aligned}$$

When $d_{\min} \to 0$ or $\beta_s \to \infty$, we have

$$2 - \frac{d_{\min} P_{\min}(w+1)}{\beta_s + d_{\min} P_{\min}} \to 2 - \frac{d_{\min} P_{\min} w}{\beta_s}.$$

Combining the above two cases establishes the lower bound for problem $\mathbf{CP_i}$.



(a) $\delta + w \geq \beta_s/(d_{\min} P_{\min})$ (b) $\delta + w < \beta_s/(d_{\min} P_{\min})$

Figure B.1: Worst case examples.

For problem **CP**, consider the case that $d_t(0) = 0$ and $a(t) \in [0,1]$, $\forall t$. In this case, it is straightforward that $\mathbf{CP_1}$ is equivalent to **CP**. Thus, the lower bound for $\mathbf{CP_i}$ is also a lower bound for **CP**. □

Theorem 4 follows from lemmas 6 and 7.

B.3 Proof of Theorem 5

Similar as the proof of Theorem 4, we only need to focus on the idling interval I_1, $[t_1, t_2]$. $\mathbf{RGCSR_s^{(w)}}$ performs as follows: it accumulates an "idling cost" and when it is less than Λ, it keeps the server idling; otherwise, it will see whether the job will comes, *i.e.*, $a_i > 0$, before the "idling cost" reaches β_s within the look-ahead window w. If so, it keeps the server idling; else it turns off the server. Let $\bar{\boldsymbol{x}}_i$ and $\boldsymbol{x}_i^{on}$ be the solutions obtained by $\mathbf{CPOFF_s}$ (Algorithm 5) and $\mathbf{RGCSR_s^{(w)}}$ (Algorithm 2), respectively. As shown in the proof of Theorem 4, the cost of the offline optimal

$\mathbf{CPOFF_s}$ is

$$Cost_{I_1}(\bar{\boldsymbol{x}}_i) = \begin{cases} D_i, & \text{if } D_i < \beta_s, \\ \beta_s, & \text{else,} \end{cases}$$

where $D_i \triangleq \sum_{t\in I_1} d_t^i p(t)$.

According to Algorithm 2, when $D_i < \alpha_s\beta_s$, we have

$$\mathrm{E}[Cost_{I_1}(\bar{\boldsymbol{x}}_i)] = D_i;$$

when $\alpha_s\beta_s \leq D_i < \beta_s$, we have

$$\mathrm{E}[Cost_{I_1}(\boldsymbol{x_i^{on}})] \leq \int_0^{D_i-\alpha_s\beta_s} (\beta_s + \Lambda) f_\Lambda(\lambda) d\Lambda + \int_{D_i-\alpha_s\beta_s}^{(1-\alpha_s)\beta_s} D_i f_\Lambda(\lambda) d\Lambda;$$

when $D_i \geq \beta_s$, we have

$$\mathrm{E}[Cost_{I_1}(\boldsymbol{x_i^{on}})] \leq \int_0^{(1-\alpha_s)\beta_s} (\beta_s + \Lambda) f_\Lambda(\lambda) d\Lambda;$$

According to PDF $f_\Lambda(\lambda)$ (Eqn. (4.2)), we can calculate $\mathrm{E}[Cost_{I_1}(\boldsymbol{x_i^{on}})]$ and the ratio between $\mathrm{E}[Cost_{I_1}(\boldsymbol{x_i^{on}})]$ and $\mathrm{E}[Cost_{I_1}(\bar{\boldsymbol{x}}_i)]$:

$$\frac{\mathrm{E}[Cost_{I_1}(\boldsymbol{x_i^{on}})]}{Cost_{I_1}(\bar{\boldsymbol{x}}_i)} \leq \begin{cases} 1, & \text{if } D_i < \alpha_s\beta_s, \\ \frac{e}{e-1+\alpha_s}, & \text{else.} \end{cases}$$

So

$$\frac{\mathrm{E}[\mathrm{C}_{\mathrm{CP_i}}(\boldsymbol{x_i^{on}})]}{\mathrm{C}_{\mathrm{CP_i}}(\bar{\boldsymbol{x}}_i)} \leq \frac{\mathrm{E}[Cost_{I_1}(\boldsymbol{x_i^{on}})]}{Cost_{I_1}(\bar{\boldsymbol{x}}_i)} \leq \frac{e}{e-1+\alpha_s}.$$

Hence, according to Theorem 3, $\mathbf{RGCSR^{(w)}}$ achieves the same competitive ratio $\frac{e}{e-1+\alpha_s}$ for **CP**.

Next, we prove no randomized algorithm can achieve a smaller competitive ratio. We set $d_t(x) = d_{\min}x,\ p(t) = P_{\min}, \forall t \in [1,T]$, then problem **CP** becomes the problem in [29]. According to [29], $\frac{e}{e-1+\alpha}$ is the competitive ratio lower bound for any randomized online algorithms.

□ **End of chapter.**

Appendix C

Proofs of Theorems 6, 7 and 8

C.1 Proof of Theorem 6

First, we show that the combined solution $\sum_{i=1}^{N} \bar{\boldsymbol{y}}_i$ is optimal to **EP**.

Denote $\mathrm{C_{EP}}(\boldsymbol{y})$ to be cost of **EP** of solution $\boldsymbol{y}$. Suppose that $\tilde{\boldsymbol{y}}$ is an optimal solution for **EP**. We will show that we can construct a new feasible solution $\sum_{i=1}^{N} \hat{\boldsymbol{y}}_i$ for **EP**, and a new feasible solution $\hat{\boldsymbol{y}}_i$ for each $\mathbf{EP}_{\mathrm{i}}$, such that

$$\mathrm{C_{EP}}(\tilde{\boldsymbol{y}}) = \mathrm{C_{EP}}(\sum_{i=1}^{N} \hat{\boldsymbol{y}}_i) = \sum_{i=1}^{N} \mathrm{C_{EP_i}}(\hat{\boldsymbol{y}}_i) + \sum_{t=1}^{T} p(t)\,[e(t) - NL]^+ . \tag{C.1}$$

$\bar{\boldsymbol{y}}_i$ is an optimal solution for each $\mathbf{EP}_{\mathrm{i}}$. Hence, $\mathrm{C_{EP_i}}(\hat{\boldsymbol{y}}_i) \geq \mathrm{C_{EP_i}}(\bar{\boldsymbol{y}}_i)$ for each i. Thus,

$$\begin{aligned} \mathrm{C_{EP}}(\tilde{\boldsymbol{y}}) &= \sum_{i=1}^{N} \mathrm{C_{EP_i}}(\hat{\boldsymbol{y}}_i) + \sum_{t=1}^{T} p(t)\,[e(t) - NL]^+ \\ &\geq \sum_{i=1}^{N} \mathrm{C_{EP_i}}(\bar{\boldsymbol{y}}_i) + \sum_{t=1}^{T} p(t)\,[e(t) - NL]^+ . \end{aligned} \tag{C.2}$$

Besides, we also can prove that

$$\sum_{i=1}^{N} \mathrm{C_{EP_i}}(\bar{\boldsymbol{y}}_i) + \sum_{t=1}^{T} p(t)\,[e(t) - NL]^+ \geq \mathrm{C_{EP}}(\sum_{i=1}^{N} \bar{\boldsymbol{y}}_i). \tag{C.3}$$

Hence, $\mathrm{C_{EP}}(\tilde{\boldsymbol{y}}) = \mathrm{C_{EP}}(\sum_{i=1}^{N} \bar{\boldsymbol{y}}_i)$, *i.e.*, $\sum_{i=1}^{N} \bar{\boldsymbol{y}}_i$ is an optimal solution for **EP**.

Then, we show $\mathrm{C_{EP}}(\sum_{i=1}^{N} \boldsymbol{y}_i^{on}) \leq \gamma \cdot \mathrm{C_{EP}}(\sum_{i=1}^{N} \bar{\boldsymbol{y}}_i)$.

Because $\mathrm{C_{EP_i}}(\boldsymbol{y}_i^{on}) \leq \gamma \cdot \mathrm{C_{EP_i}}(\bar{\boldsymbol{y}}_i)$ and $\bar{\boldsymbol{y}}_i$ is optimal for $\mathbf{EP}_{\mathrm{i}}$, we have $\gamma \geq 1$. According to Eqn. (C.2), we have

$$\gamma \cdot \mathrm{C_{EP}}(\tilde{\boldsymbol{y}}) \;\geq\; \sum_{i=1}^{N} \mathrm{C_{EP_i}}(\boldsymbol{y}_i^{on}) + \sum_{t=1}^{T} p(t)\,[e(t) - NL]^+ .$$

Besides, we also can prove that

$$\sum_{i=1}^{N} \mathrm{C}_{\mathrm{EP_i}}(\boldsymbol{y}_i^{on}) + \sum_{t=1}^{T} p(t)\,[e(t) - NL]^+ \geq \mathrm{C}_{\mathrm{EP}}(\sum_{i=1}^{N} \boldsymbol{y}_i^{on}). \tag{C.4}$$

Hence, $\mathrm{C}_{\mathrm{EP}}(\sum_{i=1}^{N} \boldsymbol{y}_i^{on}) \leq \gamma \cdot \mathrm{C}_{\mathrm{EP}}(\sum_{i=1}^{N} \bar{\boldsymbol{y}}_i)$.

It remains to prove Eqn. (C.1), (C.3) and (C.4), which we show in Lemmas 8 and 9.

Lemma 8. $\mathrm{C}_{\mathrm{EP}}(\tilde{\boldsymbol{y}}) = \mathrm{C}_{\mathrm{EP}}(\sum_{i=1}^{N} \hat{\boldsymbol{y}}_i) = \sum_{i=1}^{N} \mathrm{C}_{\mathrm{EP_i}}(\hat{\boldsymbol{y}}_i) + \sum_{t=1}^{T} p(t)\,[e(t) - NL]^+$.

Proof. Define $\hat{\boldsymbol{y}}_i$ based on $\tilde{\boldsymbol{y}}$ by:

$$\widehat{y}_i(t) = \begin{cases} 1, & \text{if } i \leq \tilde{y}(t) \\ 0, & \text{otherwise.} \end{cases} \tag{C.5}$$

It is straightforward to see that

$$\tilde{y}(t) = \sum_{i=1}^{N} \hat{y}_i(t). \tag{C.6}$$

So we have $\mathrm{C}_{\mathrm{EP}}(\tilde{\boldsymbol{y}}) = \mathrm{C}_{\mathrm{EP}}(\sum_{i=1}^{N} \hat{\boldsymbol{y}}_i)$.

According to **EP**,

$$\begin{aligned} \mathrm{C}_{\mathrm{EP}}(\sum_{i=1}^{N} \hat{\boldsymbol{y}}_i) \;&=\; \sum_{t=1}^{T} \Bigg\{ \psi\left(\sum_{i=1}^{N} \hat{y}_i(t), p(t), e(t)\right) \\ &\quad + \beta_g [\sum_{i=1}^{N} \hat{y}_i(t) - \sum_{i=1}^{N} \hat{y}_i(t-1)]^+ \Bigg\}, \end{aligned}$$

and

$$\begin{aligned} \sum_{i=1}^{N} \mathrm{C}_{\mathrm{EP_i}}(\hat{\boldsymbol{y}}_i) \;&=\; \sum_{t=1}^{T} \Bigg\{ \sum_{i=1}^{N} \psi\left(\hat{y}_i(t), p(t), e_i(t)\right) \\ &\quad + \beta_g \sum_{i=1}^{N} [\hat{y}_i(t) - \hat{y}_i(t-1)]^+ \Bigg\}. \end{aligned}$$

Note that $\widehat{y}_1(t) \geq ... \geq \widehat{y}_N(t)$ is a decreasing sequence. Because $\widehat{y}_i(t) \in \{0,1\}$, $\forall i, t$, we obtain

$$\begin{aligned}
& \sum_{i=1}^{N} [\hat{y}_i(t) - \hat{y}_i(t-1)]^+ \\
= & \begin{cases} 0, & \text{if } \sum_{i=1}^{N} \hat{y}_i(t) \leq \sum_{i=1}^{N} \hat{y}_i(t-1) \\ \sum_{i=1}^{N} \hat{y}_i(t) - \sum_{i=1}^{N} \hat{y}_i(t-1), & \text{otherwise} \end{cases} \\
= & \Big[\sum_{i=1}^{N} \hat{y}_i(t) - \sum_{i=1}^{N} \hat{y}_i(t-1) \Big]^+ . \qquad \text{(C.7)}
\end{aligned}$$

Also, according to Eqn. (2.9), $\psi\left(y(t), p(t), e(t)\right)$ can be rewritten as:

$$\begin{aligned}
& \psi\left(y(t), p(t), e(t)\right) \\
\triangleq & \begin{cases} c_m y(t) + p(t)e(t), & \text{if } p(t) \leq c_o, \\ c_m y(t) + p(t)e(t) + & \text{else.} \\ [c_o - p(t)] \min\{e(t), Ly(t)\} & \end{cases} \qquad \text{(C.8)}
\end{aligned}$$

Next, we distinguish two cases:

Case 1: $e(t) < NL$. In this case, $\sum_{i=1}^{N} e_i(t) = e(t)$ and $[e(t) - NL]^+ = 0$. According to the definition of $e_i(t)$, denoting $\bar{N} = \lfloor e(t)/L \rfloor < N$, we have

$$e_i(t) = \begin{cases} L, & \text{if } i \leq \bar{N}, \\ e(t) - \bar{N}L, & \text{if } i = \bar{N} + 1, \\ 0, & \text{else.} \end{cases}$$

Because $\widehat{y}_1(t) \geq ... \geq \widehat{y}_N(t)$ is a decreasing sequence and $\widehat{y}_i(t) \in \{0,1\}$, $\forall t$, we have

$$\begin{aligned}
\sum_{i=1}^{N} \min\{e_i(t), L\hat{y}_i(t)\} & = \begin{cases} L \sum_{i=1}^{N} \hat{y}_i(t), & \text{if } \sum_{i=1}^{N} \hat{y}_i(t) \leq \bar{N}, \\ e(t) & \text{else.} \end{cases} \\
& = \min\{e(t), L \sum_{i=1}^{N} \hat{y}_i(t)\}.
\end{aligned}$$

Thus, by Eqn. (C.8), we have

$$\psi\left(\sum_{i=1}^{N}\hat{y}_i(t), p(t), e(t)\right) = \sum_{i=1}^{N}\psi\left(\hat{y}_i(t), p(t), e_i(t)\right) + p(t)\left[e(t) - NL\right]^+ . \tag{C.9}$$

Case 2: $e(t) \geq NL$. In this case, $e_i(t) = L$, $\forall i \in [1, N]$, we have

$$\sum_{i=1}^{N}\min\{e_i(t), L\hat{y}_i(t)\} = L\sum_{i=1}^{N}\hat{y}_i(t) = \min\{e(t), L\sum_{i=1}^{N}\hat{y}_i(t)\}.$$

Thus, by Eqn. (C.8), we have

$$\psi\left(\sum_{i=1}^{N}\hat{y}_i(t), p(t), e(t)\right) = \sum_{i=1}^{N}\psi\left(\hat{y}_i(t), p(t), e_i(t)\right) + p(t)\left[e(t) - NL\right]^+ . \tag{C.10}$$

By Eqns. (C.7), (C.9) and (C.10), we have $\mathrm{C_{EP}}(\sum_{i=1}^{N}\hat{\boldsymbol{y}}_i) = \sum_{i=1}^{N}\mathrm{C_{EP_i}}(\hat{\boldsymbol{y}}_i) + \sum_{t=1}^{T}p(t)\left[e(t) - NL\right]^+$.

This completes the proof of this lemma. □

Lemma 9. $\sum_{i=1}^{N}\mathrm{C_{EP_i}}(\boldsymbol{y}_i) + \sum_{t=1}^{T}p(t)\left[e(t) - NL\right]^+ \geq$ $\mathrm{C_{EP}}(\sum_{i=1}^{N}\boldsymbol{y}_i)$, where $\boldsymbol{y}_i$ is any feasible solution for problem $\boldsymbol{EP}_\mathrm{i}$

Proof. First, it is straightforward that

$$\sum_{i=1}^{N}[y_i(t) - y_i(t-1)]^+ \geq \Big[\sum_{i=1}^{N}y_i(t) - \sum_{i=1}^{N}y_i(t-1)\Big]^+. \tag{C.11}$$

Then by Eqn. (C.8) and the fact that $\sum_{i=1}^{N}e_i(t) = \min\{e(t), NL\}$ and

$$\begin{aligned}\sum_{i=1}^{N}\min\{e_i(t), Ly_i(t)\} &\leq \min\{\sum_{i=1}^{N}e_i(t), L\sum_{i=1}^{N}y_i(t)\}\\ &\leq \min\{e(t), L\sum_{i=1}^{N}y_i(t)\},\end{aligned}$$

we have

$$\psi\left(\sum_{i=1}^{N}\bar{y}_i(t), p(t), e(t)\right) \leq \sum_{i=1}^{N}\psi\left(\bar{y}_i(t), p(t), e_i(t)\right) + p(t)\left[e(t) - NL\right]^+ . \quad \text{(C.12)}$$

This lemma follows from Eqns. (C.11) and (C.12). □

C.2 Proof of Theorem 7

Instead of proving this theorem directly, we prove a stronger theorem that fully characterizes an offline optimal solution. Then Theorem 7 follows naturally. An very important structure of an offline optimal solution is "critical segments", which are constructed according to $R_i(t)$.

Definition 1. *We divide all time intervals in* $[1, T]$ *into disjoint parts called* critical segments*:*

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

The critical segments are characterized by a set of critical points*:* $T_1^c < T_2^c < ... < T_k^c$*. We define each critical point* T_j^c *along with an auxiliary point* $\tilde{T}_j^c$*, such that the pair* $(T_j^c, \tilde{T}_j^c)$ *satisfy the following conditions:*

(Boundary): Either $\left(R_i(T_j^c) = 0 \text{ and } R_i(\tilde{T}_j^c) = -\beta_g\right)$ *or* $\left(R_i(T_j^c) = -\beta_g \text{ and } R_i(\tilde{T}_j^c) = 0\right)$*.*

(Interior): $-\beta < R_i(\tau) < 0$ *for all* $T_j^c < \tau < \tilde{T}_j^c$*.*

In other words, each pair of $(T_j^c, \tilde{T}_j^c)$ corresponds to an interval where $R_i(t)$ goes from -β_g to 0 or 0 to -β_g, without reaching the two extreme values inside the interval. For example, $(T_1^c, \tilde{T}_1^c)$ and $(T_2^c, \tilde{T}_2^c)$ in Fig. C.1 are two such pairs, while the corresponding critical segments are (T_1^c, T_2^c) and (T_2^c, T_3^c). It is straightforward to see that all $(T_j^c, \tilde{T}_j^c)$ are uniquely defined, and hence critical segments are well-defined. See Fig. C.1 for an example.

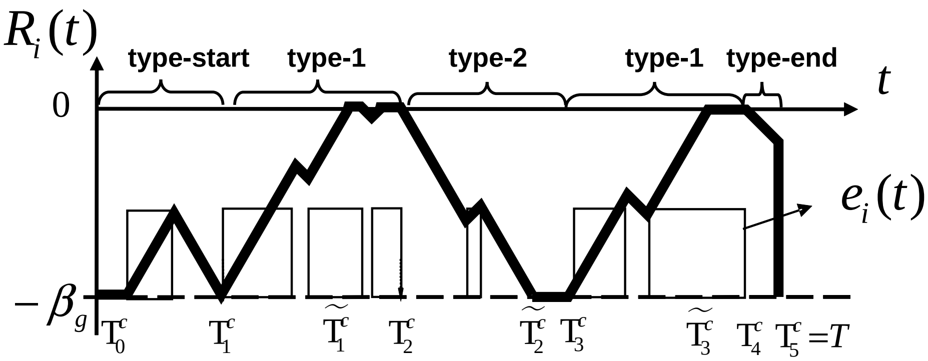


Figure C.1: An example of critical segments.

Once the time horizon $[1, T]$ is divided into critical segments, we can now characterize the optimal solution.

Definition 2. *We classify the* type *of a critical segment by:*

Type-start *(also call* type-0*):* $[1, T_1^c]$

Type-1*:* $[T_j^c + 1, T_{j+1}^c]$*, if* $R_i(T_j^c) = -\beta_g$ *and* $R_i(T_{j+1}^c) = 0$

Type-2*:* $[T_j^c + 1, T_{j+1}^c]$*, if* $R_i(T_j^c) = 0$ *and* $R_i(T_{j+1}^c) = -\beta_g$

Type-end *(also call* type-3*):* $[T_k^c + 1, T]$

For completeness, we also let $T_0^c = 0$ and $T_{k+1}^c = T$.

Then the following theorem characterizes an offline optimal solution.

Theorem 10. *An optimal solution for* $\boldsymbol{EP_i}$ *is given by*

$$y_{\text{OFA}}(t) \triangleq \begin{cases} 0, & \textit{if } t \in [T_j^c + 1, T_{j+1}^c] \textit{ is type-start/-2/-end,} \\ 1, & \textit{if } t \in [T_j^c + 1, T_{j+1}^c] \textit{ is type-1.} \end{cases} \tag{C.13}$$

Theorem 7 follows from Theorem 10 and Definition 2. Thus, it remains to prove Theorem 10.

Proof of Theorem 10:

Before we prove the theorem, we introduce a lemma.

We define the cost with regard to a segment j by:

$$\begin{aligned}&\mathrm{C}_{\mathrm{EP_i^{sg-j}}}(y)\\ \triangleq &\sum_{t=T_j^c+1}^{T_{j+1}^c} \psi\left(y(t), p(t), e_i(t)\right) + \sum_{t=T_j^c+1}^{T_{j+1}^c+1} \beta_g \cdot [y(t) - y(t-1)]^+\end{aligned}$$

and define a subproblem for critical segment j by:

$$\begin{aligned}\mathbf{EP_i}^{\mathbf{sg\text{-}j}}(\mathrm{y}_j^l, \mathrm{y}_j^r) : \min\ & \mathrm{C}_{\mathrm{EP_i^{sg-j}}}(y)\\ \text{s.t. } & y(T_j^c) = \mathrm{y}_j^l,\ y(T_{j+1}^c + 1) = \mathrm{y}_j^r,\\ \text{var } & y(t) \in \{0, 1\}, t \in [T_j^c + 1, T_{j+1}^c].\end{aligned}$$

Note that due to the startup cost across segment boundaries, in general $\mathrm{C}_{\mathrm{EP_i}} \neq \sum \mathrm{C}_{\mathrm{EP_i^{sg-j}}}(y)$. In other words, we should not expect that putting together the solutions to each segment will lead to an overall offline optimal solution. However, the following lemma shows an important structure property that one optimal solution of $\mathbf{EP_i^{sg-j}}(y_j^l, y_j^r)$ is independent of boundary conditions (y_j^l, y_j^r) although the optimal value depends on boundary conditions.

Lemma 10. $(y_{\mathrm{OFA}}(t))_{t=T_j^c+1}^{T_{j+1}^c}$ *in (C.13) is an optimal solution for* $\mathbf{EP_i^{sg-j}}(y_j^l, y_j^r)$, *despite any boundary conditions* $(\mathrm{y}_j^l, \mathrm{y}_j^r)$.

We first use this lemma to prove Theorem 10 and then we prove this lemma. Suppose $(y^*(t))_{t=1}^T$ is an optimal solution for $\mathbf{EP}_{\mathrm{i}}$. For completeness, we let $y^*(0) = 0$ and $y^*(T + 1) = 0$. We define a sequence $(y_0(t))_{t=1}^T, (y_1(t))_{t=1}^T, ..., (y_{k+1}(t))_{t=1}^T$ as follows:

1. $y_0(t) = y^*(t)$ for all $t \in [1, T]$.

2. For all $t \in [1, T]$ and $j = 1, ..., k$

$$y_j(t) = \begin{cases} y_{\mathrm{OFA}}(t), & \text{if } t \in [1, T_j^c] \\ y^*(t), & \text{otherwise} \end{cases} \tag{C.14}$$

3. $y_{k+1}(t) = y_{\text{OFA}}(t)$ for all $t \in [1, T]$.

We next set the boundary conditions for each $\mathbf{EP_i^{sg-j}}$ by

$$\mathrm{y}_j^l = y_{\text{OFA}}(T_j^c) \text{ and } \mathrm{y}_j^r = y^*(T_{j+1}^c + 1) \tag{C.15}$$

It follows that

$$\mathrm{C}_{\mathrm{EP_i}}(y_j) - \mathrm{C}_{\mathrm{EP_i}}(y_{j+1}) = \mathrm{C}_{\mathrm{EP_i^{sg-j}}}(y^*) - \mathrm{C}_{\mathrm{EP_i^{sg-j}}}(y_{\text{OFA}}) \tag{C.16}$$

By Lemma 10, we obtain $\mathrm{C}_{\mathrm{EP_i^{sg-j}}}(y^*) \geq \mathrm{C}_{\mathrm{EP_i^{sg-j}}}(y_{\text{OFA}})$ for all j. Hence,

$$\mathrm{C}_{\mathrm{EP_i}}(y^*) = \mathrm{C}_{\mathrm{EP_i}}(y_0) \geq ... \geq \mathrm{C}_{\mathrm{EP_i}}(y_{k+1}) = \mathrm{C}_{\mathrm{EP_i}}(y_{\text{OFA}}) \tag{C.17}$$

This completes the proof of Theorem 10.

Proof of Lemma 10: Consider given any boundary condition $(\mathrm{y}_j^l, \mathrm{y}_j^r)$ for $\mathbf{EP_i^{sg-j}}$. Suppose $(\widehat{y}(t))_{t=T_j^c+1}^{T_{j+1}^c}$ is an optimal solution for $\mathbf{EP_i^{sg-j}}$ w.r.t. $(\mathrm{y}_j^l, \mathrm{y}_j^r)$, and $\widehat{y} \neq y_{\text{OFA}}$. We aim to show $\mathrm{C}_{\mathrm{EP_i^{sg-j}}}(\widehat{y}) \geq \mathrm{C}_{\mathrm{EP_i^{sg-j}}}(y_{\text{OFA}})$, by considering the types of critical segment.

(**type-1**): First, suppose that critical segment $[T_j^c + 1, T_{j+1}^c]$ is type-1. Hence, $y_{\text{OFA}}(t) = 1$ for all $t \in [T_j^c + 1, T_{j+1}^c]$. Hence,

$$\mathrm{C}_{\mathrm{EP_i^{sg-j}}}(y_{\text{OFA}}) = \beta_g \cdot (1 - \mathrm{y}_j^l) + \sum_{t=T_j^c+1}^{T_{j+1}^c} \psi\big(1, p(t), e_i(t)\big) \tag{C.18}$$

Case 1: Suppose $\widehat{y}(t) = 0$ for all $t \in [T_j^c + 1, T_{j+1}^c]$. Hence,

$$\mathrm{C}_{\mathrm{EP_i^{sg-j}}}(\widehat{y}) = \beta_g \cdot \mathrm{y}_j^r + \sum_{t=T_j^c+1}^{T_{j+1}^c} \psi\big(0, p(t), e_i(t)\big) \tag{C.19}$$

We obtain:

$$\begin{aligned}
& \mathrm{C}_{\mathrm{EP_i^{sg-j}}}(\widehat{y}) - \mathrm{C}_{\mathrm{EP_i^{sg-j}}}(y_{\text{OFA}}) \\
= \ & \beta_g \cdot \mathrm{y}_j^r + \sum_{t=T_j^c+1}^{T_{j+1}^c} r_i(t) - \beta_g(1 - \mathrm{y}_j^l) & \text{(C.20)} \\
\geq \ & \beta_g \cdot \mathrm{y}_j^r + R_i(T_{j+1}^c) - R_i(T_j^c) - \beta_g(1 - \mathrm{y}_j^l) & \text{(C.21)} \\
= \ & \beta_g \cdot \mathrm{y}_j^r + \beta_g - \beta_g + \beta_g \mathrm{y}_j^l \geq 0 & \text{(C.22)}
\end{aligned}$$

where Eqn. (C.20) follows from the definition of $r_i(t)$ (see Eqn. (5.3)) and Eqn. (C.21) follows from Lemma 11. This completes the proof for Case 1.

(**Case 2**): Suppose $\widehat{y}(t) = 1$ for some $t \in [T_j^c + 1, T_{j+1}^c]$. This implies that $\mathrm{C}_{\mathrm{EP}_{\mathrm{i}}^{\mathrm{sg-j}}}(\widehat{y})$ has to involve the startup cost β_g.

Next, we denote the minimal set of segments within $[T_j^c + 1, T_{j+1}^c]$ by

$$[\tau_1^b, \tau_1^e], [\tau_2^b, \tau_2^e], [\tau_3^b, \tau_3^e], ..., [\tau_p^b, \tau_p^e]$$

such that $\widehat{y}(t) \neq y_{\mathrm{OFA}}(t)$ for all $t \in [\tau_l^b, \tau_l^e]$, $l \in \{1, ..., p\}$, where $\tau_l^e < \tau_{l+1}^b$.

Since $\widehat{y} \neq y_{\mathrm{OFA}}$, then there exists at least one $t \in [T_j^c+1, T_{j+1}^c]$ such that $\widehat{y}(t) = 0$. Hence, τ_1^b is well-defined.

Note that upon exiting each segment $[\tau_l^b, \tau_l^e]$, $\widehat{y}$ switches from 0 to 1. Hence, it incurs the startup cost β_g. However, when $\tau_p^e = T_{j+1}^c$ and $\mathrm{y}_j^r = 0$, the startup cost is not for critical segment $[T_j^c + 1, T_{j+1}^c]$.

Therefore, we obtain:

$$\begin{aligned}
& \mathrm{C}_{\mathrm{EP}_{\mathrm{i}}^{\mathrm{sg-j}}}(\widehat{y}) - \mathrm{C}_{\mathrm{EP}_{\mathrm{i}}^{\mathrm{sg-j}}}(y_{\mathrm{OFA}}) && \text{(C.23)} \\
= & \sum_{t=\tau_1^b}^{\tau_1^e} r_i(t) + \beta_g \cdot \mathbf{1}[\tau_1^b \neq T_j^c + 1] && \text{(C.24)} \\
& + \sum_{l=2}^{p-1} \Big(\sum_{t=\tau_l^b}^{\tau_l^e} r_i(t) + \beta_g \Big) && \text{(C.25)} \\
& + \sum_{t=\tau_p^b}^{\tau_p^e} r_i(t) + \beta_g \mathrm{y}_j^r \cdot \mathbf{1}[\tau_p^e = T_{j+1}^c] + \beta_g \cdot \mathbf{1}[\tau_p^e \neq T_{j+1}^c]. && \text{(C.26)}
\end{aligned}$$

Now we prove the terms (C.24) (C.25) and (C.26) are all no less than 0.

First, if $\tau_1^b = T_j^c + 1$, then

$$\begin{aligned}
\sum_{t=\tau_1^b}^{\tau_1^e} r_i(t) + \beta_g \cdot \mathbf{1}[\tau_1^b \neq T_j^c + 1] &= \sum_{t=T_j^c+1}^{\tau_1^e} r_i(t) \\
&\geq R_i(\tau_1^e) - R_i(T_j^c) \\
&\geq R_i(\tau_1^e) + \beta_g \geq 0.
\end{aligned}$$

else then

$$\begin{aligned}\sum_{t=\tau_1^b}^{\tau_1^e} r_i(t) + \beta_g \cdot \mathbf{1}[\tau_1^b \neq T_j^c + 1] &= \sum_{t=\tau_1^b}^{\tau_1^e} r_i(t) + \beta_g \\ &\geq R_i(\tau_1^e) - R_i(\tau_1^b - 1) + \beta_g \\ &\geq R_i(\tau_1^e) + \beta_g \geq 0.\end{aligned}$$

Thus, we proved (C.24)≥ 0.

Second,

$$\begin{aligned}\sum_{t=\tau_l^b}^{\tau_l^e} r_i(t) + \beta_g &\geq R_i(\tau_l^e) - R_i(\tau_l^b - 1) + \beta_g \\ &\geq R_i(\tau_l^e) + \beta_g \geq 0.\end{aligned}$$

Thus, we proved (C.25)≥ 0.

Last, if $\tau_p^e = T_{j+1}^c$, then

$$\begin{aligned}&\sum_{t=\tau_p^b}^{\tau_p^e} r_i(t) + \beta_g \mathrm{y}_i^r \cdot \mathbf{1}[\tau_p^e = T_{j+1}^c] + \beta_g \cdot \mathbf{1}[\tau_p^e \neq T_{j+1}^c] \\ \geq\ & \sum_{t=\tau_p^b}^{T_{j+1}^c} r_i(t) \geq R_i(T_{j+1}^c) - R_i(\tau_p^b - 1) \\ =\ & -R_i(\tau_p^b - 1) \geq 0.\end{aligned}$$

else then

$$\begin{aligned}&\sum_{t=\tau_p^b}^{\tau_p^e} r_i(t) + \beta_g \mathrm{y}_i^r \cdot \mathbf{1}[\tau_p^e = T_{j+1}^c] + \beta_g \cdot \mathbf{1}[\tau_p^e \neq T_{j+1}^c] \\ =\ & \sum_{t=\tau_p^b}^{\tau_p^e} r_i(t) + \beta_g \geq R_i(\tau_p^e) - R_i(\tau_p^b - 1) + \beta_g \\ \geq\ & 0.\end{aligned}$$

Thus, we proved (C.26)≥ 0.

So we obtain

$$\mathrm{C}_{\mathrm{EP}_{\mathrm{i}}^{\mathrm{sg-j}}}(\widehat{y}) - \mathrm{C}_{\mathrm{EP}_{\mathrm{i}}^{\mathrm{sg-j}}}(y_{\mathrm{OFA}}) \geq 0.$$

(**type-2**): Next, suppose that critical segment $[T_j^c+1, T_{j+1}^c]$ is type-2. Hence, $y_{\mathrm{OFA}}(t) = 0$ for all $t \in [T_j^c+1, T_{j+1}^c]$. Note that the above argument applies similarly to type-2 setting, when we consider (Case 1): $\widehat{y}(t) = 1$ for all $t \in [T_j^c+1, T_{j+1}^c]$ and (Case 2): $\widehat{y}(t) = 0$ for some $t \in [T_j^c+1, T_{j+1}^c]$.

(**type-start** and **type-end**): We note that the argument of type-2 applies similarly to type-start and type-end settings.

Therefore, we complete the proof by showing $\mathrm{C}_{\mathrm{EP}_{\mathrm{i}}^{\mathrm{sg-j}}}(\widehat{y}) \geq \mathrm{C}_{\mathrm{EP}_{\mathrm{i}}^{\mathrm{sg-j}}}(y_{\mathrm{OFA}})$ for all $j \in [0, k]$.

Lemma 11. *Suppose $\tau_1, \tau_2 \in [T_j^c+1, T_{j+1}^c]$ and $\tau_1 < \tau_2$. Then,*

$$R_i(\tau_2) - R_i(\tau_1) \begin{cases} \leq \sum_{t=\tau_1+1}^{\tau_2} r_i(t), & \text{if } [T_j^c+1, T_{j+1}^c] \text{ is type-1} \\ \geq \sum_{t=\tau_1+1}^{\tau_2} r_i(t), & \text{if } [T_j^c+1, T_{j+1}^c] \text{ is type-2} \end{cases} \tag{C.27}$$

Proof. We recall that

$$R_i(t) \triangleq \min\Big\{0, \max\{-\beta_g, R_i(t-1) + r_i(t)\}\Big\} \tag{C.28}$$

First, we consider $[T_j^c+1, T_{j+1}^c]$ as type-1. This implies that only $R_i(T_j^c) = -\beta_g$, whereas $R_i(t) > -\beta_g$ for $t \in [T_j^c+1, T_{j+1}^c]$. Hence,

$$R_i(t) = \min\{0, R_i(t-1) + r_i(t)\} \leq R_i(t-1) + r_i(t) \tag{C.29}$$

Iteratively, we obtain

$$R_i(\tau_2) \leq R_i(\tau_1) + \sum_{t=\tau_1+1}^{\tau_2} r_i(t) \tag{C.30}$$

When $[T_j^c+1, T_{j+1}^c]$ is type-2, we proceed with a similar proof, except

$$R_i(t) = \max\{-\beta_g, R_i(t-1) + r_i(t)\} \geq R_i(t-1) + r_i(t) \tag{C.31}$$

Therefore,

$$R_i(\tau_2) \geq R_i(\tau_1) + \sum_{t=\tau_1+1}^{\tau_2} r_i(t). \tag{C.32}$$

□

C.3 Proof of Theorem 8

First, we focus on a sub-problem $\mathbf{EP}_\mathrm{i}$ with energy demand $\boldsymbol{e}_i$. We denote the set of indexes of critical segments for type-h by $\mathcal{T}_h \subseteq \{0, .., k\}$. Note that we also refer to type-start and type-end by type-0 and type-3 respectively.

Define the sub-cost for type-h by

$$\mathrm{C}^{\text{ty-h}}_{\mathrm{EP_i}}(y) \triangleq \sum_{j\in\mathcal{T}_h} \sum_{t=T^c_j+1}^{T^c_{j+1}} \psi\left(y(t), p(t), e_i(t)\right) + \beta_g \cdot [y(t) - y(t-1)]^+.$$

Hence, $\mathrm{C}_{\mathrm{EP_i}}(y) = \sum_{h=0}^{3} \mathrm{C}^{\text{ty-h}}_{\mathrm{EP_i}}(y)$. We prove by comparing the sub-cost for each type-h. We denote the outcome of $\mathbf{CHASE_s^{(w)}}$ by $\left(y_{\mathrm{CHASE(w)}}(t)\right)_{t=1}^{T}$.

(**type-0**): Note that both $y_{\mathrm{OFA}}(t) = y_{\mathrm{CHASE(w)}}(t) = 0$ for all $t \in [1, T^c_1]$. Hence,

$$\mathrm{C}^{\text{ty-0}}_{\mathrm{EP_i}}(y_{\mathrm{OFA}}) = \mathrm{C}^{\text{ty-0}}_{\mathrm{EP_i}}(y_{\mathrm{CHASE(w)}}).$$

(**type-1**): Based on the definition of critical segment (Definition 1), we recall that there is an auxiliary point $\tilde{T}^c_j$, such that either $\left(R_i(T^c_j) = 0 \text{ and } R_i(\tilde{T}^c_j) = -\beta_g\right)$ or $\left(R_i(T^c_j) = -\beta_g \text{ and } R_i(\tilde{T}^c_j) = 0\right)$. We focus on the segment $T^c_j + 1 + w < \tilde{T}^c_j$. We observe

$$y_{\mathrm{CHASE(w)}}(t) = \begin{cases} 0, & \text{for all } t \in [T^c_j + 1, \tilde{T}^c_j - w), \\ 1, & \text{for all } t \in [\tilde{T}^c_j - w, T^c_{j+1}]. \end{cases}$$

We consider a particular type-1 critical segment, i.e., k-th type-1 critical segment: $[T^c_j + 1, T^c_{j+1}]$. Note that by the definition of type-1, $y_{\mathrm{OFA}}(T^c_j) = y_{\mathrm{CHASE(w)}}(T^c_j) = 0$. $y_{\mathrm{OFA}}(t)$ switches from 0 to 1 at time $T^c_j + 1$, while $y_{\mathrm{CHASE(w)}}$ switches at time $\tilde{T}^c_j - w$, both incurring startup cost β_g. The cost difference between $y_{\mathrm{CHASE(w)}}$ and y_{OFA}

within $[T_j^c+1, T_{j+1}^c]$ is

$$\begin{aligned}
&\sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1}\Big(\psi\left(0,p(t),e_i(t)\right)-\psi\Big(1,\sigma(t),e_i(t)\Big)\Big)+\beta_g-\beta_g\\
=&\sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1} r_i(t)=R_i(\tilde{T}_j^c-w-1)-R_i(T_j^c)=q_k^1+\beta_g,
\end{aligned}$$

where $q_k^1 \triangleq R_i(\tilde{T}_j^c-w-1)$.

Recall the number of type-h critical segments $m_h \triangleq |\mathcal{T}_h|$.

$$\mathrm{C}_{\mathrm{EP_i}}^{\text{ty-1}}(y_{\mathrm{CHASE(w)}}) \le \mathrm{C}_{\mathrm{EP_i}}^{\text{ty-1}}(y_{\mathrm{OFA}})+m_1\cdot\beta_g+\sum_{k=1}^{m_1} q_k^1.$$

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

$$\begin{aligned}
\mathrm{C}_{\mathrm{EP_i}}^{\text{ty-h}}(y_{\mathrm{CHASE(w)}}) &\le \mathrm{C}_{\mathrm{EP_i}}^{\text{ty-h}}(y_{\mathrm{OFA}})-\sum_{k=1}^{m_h} q_k^h\\
&\le \mathrm{C}_{\mathrm{EP_i}}^{\text{ty-h}}(y_{\mathrm{OFA}})+\beta_g m_h.
\end{aligned}$$

The last inequality comes from that $q_k^h \ge -\beta_g$ for all h,k.

Furthermore, we note $m_1=m_2+m_3$. Overall, we obtain

$$\begin{aligned}
&\frac{\mathrm{C}_{\mathrm{EP_i}}(y_{\mathrm{CHASE(w)}})}{\mathrm{C}_{\mathrm{EP_i}}(y_{\mathrm{OFA}})}=\frac{\sum_{h=0}^{3}\mathrm{C}_{\mathrm{EP_i}}^{\text{ty-h}}(y_{\mathrm{CHASE(w)}})}{\sum_{h=0}^{3}\mathrm{C}_{\mathrm{EP_i}}^{\text{ty-h}}(y_{\mathrm{OFA}})}\\
\le&\ \frac{m_1\beta_g+\sum_{k=1}^{m_1}q_k^1+(m_2+m_3)\beta_g+\sum_{h=0}^{3}\mathrm{C}_{\mathrm{EP_i}}^{\text{ty-h}}(y_{\mathrm{OFA}})}{\sum_{h=0}^{3}\mathrm{C}_{\mathrm{EP_i}}^{\text{ty-h}}(y_{\mathrm{OFA}})}\\
=&\ 1+\frac{2m_1\beta_g+\sum_{k=1}^{m_1}q_k^1}{\sum_{h=0}^{3}\mathrm{C}_{\mathrm{EP_i}}^{\text{ty-h}}(y_{\mathrm{OFA}})}\\
\le&\ 1+\begin{cases}0 & \text{if } m_1=0,\\ \dfrac{2m_1\beta_g+\sum_{k=1}^{m_1}q_k^1}{\mathrm{C}_{\mathrm{EP_i}}^{\text{ty-1}}(y_{\mathrm{OFA}})} & \text{otherwise.}\end{cases}
\end{aligned}$$

By Lemma 12 and simplifications, we obtain

$$\begin{aligned}
&\frac{\mathrm{C}_{\mathrm{EP_i}}(y_{\mathrm{CHASE(w)}})}{\mathrm{C}_{\mathrm{EP_i}}(y_{\mathrm{OFA}})} \\
\leq\ & 1+\frac{2\beta_g\big(LP_{\max}-Lc_o-c_m\big)}{\beta_g LP_{\max}+w\cdot c_m P_{\max}\big(L-\frac{c_m}{P_{\max}-c_o}\big)\big)} \\
\leq\ & 1+\frac{2\big(P_{\max}-c_o\big)}{P_{\max}(1+wc_m/\beta_g)}. \qquad \text{(C.33)}
\end{aligned}$$

Hence, according to Theorem 6, $\mathbf{CHASE^{(w)}}$ achieves the same competitive ratio for problem **EP**.

Lemma 12.

$$\begin{aligned}
\mathrm{C}^{\text{ty-1}}_{\mathrm{EP_i}}(y_{\mathrm{OFA}}) \ \geq\ & m_1\beta_g+\sum_{k=1}^{m_1}\Big(\frac{(q_k^1+\beta_g)(Lc_o+c_m)}{LP_{\max}-Lc_o-c_m} \\
& +w\cdot c_m+\frac{c_o(-q_k^1+w\cdot c_m)}{P_{\max}-c_o}\Big) \\
\geq\ & \frac{m_1 P_{\max}(\beta_g+wc_m)}{P_{\max}-c_o}.
\end{aligned}$$

Proof. Consider a particular type-1 segment $[T_j^c+1, T_{j+1}^c]$. Denote the costs of $\mathrm{y_{OFA}}$ during $[T_j^c+1, \tilde{T}_j^c-w-1]$ and $[\tilde{T}_j^c-w, T_{j+1}^c]$ by $\mathrm{Cost^{up}}$ and $\mathrm{Cost^{pt}}$ respectively.

Step 1: We bound $\mathrm{Cost^{up}}$ as follows:

$$\begin{aligned}
&\mathrm{Cost^{up}} \\
=\ & \beta_g+\sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1}\psi\left(1,p(t),e_i(t)\right) \\
=\ & \beta_g+(\tilde{T}_j^c-w-1-T_j^c)c_m+\sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1}\big(\psi\left(1,p(t),e_i(t)\right)-c_m\big). \qquad \text{(C.34)}
\end{aligned}$$

On the other hand, we obtain

$$
\begin{aligned}
& \sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1} \big(\psi\left(1, p(t), e_i(t)\right) - c_m\big) \\
= \; & \frac{\sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1} \big(\psi\left(1, p(t), e_i(t)\right) - c_m\big)}{\sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1} \big(\psi\left(0, p(t), e_i(t)\right) - \psi\left(1, p(t), e_i(t)\right) + c_m\big)} \\
& \times \sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1} \big(\psi\left(0, p(t), e_i(t)\right) - \psi\left(1, p(t), e_i(t)\right) + c_m\big) \\
\geq \; & \min_{\tau \in [T_j^c+1, \tilde{T}_j^c-w-1]} \frac{\psi\left(1, p(\tau), e_i(\tau)\right) - c_m}{\psi\left(0, p(\tau), e_i(\tau)\right) - \psi\left(1, p(\tau), e_i(\tau)\right) + c_m} \\
& \times \sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1} \big(\psi\left(0, p(t), e_i(t)\right) - \psi\left(1, p(t), e_i(t)\right) + c_m\big) \\
\geq \; & \frac{c_o}{P_{\max} - c_o} \\
& \times \sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1} \big(\psi\left(0, p(t), e_i(t)\right) - \psi\left(1, p(t), e_i(t)\right) + c_m\big).
\end{aligned}
\tag{C.35}
$$

The last inequality follows from Lemma 13.

Next, we bound the second term by

$$
\begin{aligned}
& \sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1} \big(\psi\left(0, p(t), e_i(t)\right) - \psi\left(1, p(t), e_i(t)\right) + c_m\big) \\
\geq \; & \sum_{t=T_j^c+1}^{\tilde{T}_j^c-w-1} \big(r_i(t) + c_m\big) \\
\geq \; & R_i\big(\tilde{T}_j^c - w - 1\big) - R_i\big(T_j^c\big) + (\tilde{T}_j^c - w - 1 - T_j^c)c_m \\
= \; & q_k^1 + \beta_g + (\tilde{T}_j^c - w - 1 - T_j^c)c_m.
\end{aligned}
$$

Together, we obtain

$$\begin{aligned}
& \text{Cost}^{\text{up}} \\
\geq\ & \beta_g + (\tilde{T}_j^c - w - 1 - T_j^c)c_m + \\
& \frac{c_o}{P_{\max} - c_o}\Big(q_k^1 + \beta_g + (\tilde{T}_j^c - w - 1 - T_j^c)c_m\Big) \\
=\ & \beta_g + \frac{(q_k^1 + \beta_g)c_o + (\tilde{T}_j^c - w - 1 - T_j^c)P_{\max}c_m}{P_{\max} - c_o}.
\end{aligned} \tag{C.36}$$

Furthermore, we note that $\big(\tilde{T}_j^c - w - 1 - T_j^c\big)$ is lower bounded by the steepest descend when $p(t) = P_{\max}$ and $e_i(t) = L$,

$$\tilde{T}_j^c - w - 1 - T_j^c \geq \frac{q_k^1 + \beta_g}{L\big(P_{\max} - c_o\big) - c_m} \tag{C.37}$$

By Eqns. (C.36)-(C.37), we obtain

$$\begin{aligned}
& \text{Cost}^{\text{up}} \\
\geq\ & \beta_g + \frac{(q_k^1 + \beta_g)c_o + (\tilde{T}_j^c - w - 1 - T_j^c)P_{\max}c_m}{P_{\max} - c_o} \\
\geq\ & \beta_g + \frac{(q_k^1 + \beta_g)(Lc_o + c_m)}{L\big(P_{\max} - c_o\big) - c_m}.
\end{aligned} \tag{C.38}$$

Step 2: We bound Cost^{pt} as follows.

$$\begin{aligned}
& \text{Cost}^{\text{pt}} = \sum_{t=\tilde{T}_j^c - w}^{T_{j+1}^c} \psi\left(1, p(t), e_i(t)\right) \\
=\ & (T_{j+1}^c - \tilde{T}_j^c + w + 1)c_m + \sum_{t=\tilde{T}_j^c - w}^{T_{j+1}^c} \Big(\psi\left(1, p(t), e_i(t)\right) - c_m\Big) \\
\geq\ & w \cdot c_m + \\
& \frac{c_o}{P_{\max} - c_o} \sum_{t=\tilde{T}_j^c - w}^{T_{j+1}^c} \Big(\psi\left(0, p(t), e_i(t)\right) - \psi\left(1, p(t), e_i(t)\right) + c_m\Big).
\end{aligned}$$

On the other hand, we obtain

$$\begin{aligned}
&\sum_{t=\tilde{T}_j^c-w}^{T_{j+1}^c} \big(\psi\left(0,p(t),e_i(t)\right)-\psi\left(1,p(t),e_i(t)\right)+c_m \\
= &\sum_{t=\tilde{T}_j^c-w}^{T_{j+1}^c} r_i(t)+(T_{j+1}^c-\tilde{T}_j^c+w+1)c_m \\
\geq \; & R_i(T_{j+1}^c)-R_i(\tilde{T}_j^c-w-1)+w\cdot c_m = w\cdot c_m - q_k^1.
\end{aligned}$$

Therefore,

$$\text{Cost}^{\text{pt}} \;\; \geq \;\; w\cdot c_m+\frac{c_o(w\cdot c_m-q_k^1)}{P_{\max}-c_o}. \tag{C.39}$$

Since there are m_1 type-1 critical segments, according to Eqns. (C.38)-(C.39), we obtain

$$\begin{aligned}
&\text{Cost}^{\text{ty-1}}(y_{\text{OFA}}) \\
\geq \; & m_1\beta_g+\sum_{k=1}^{m_1}\Big(\frac{(q_k^1+\beta_g)(Lc_o+c_m)}{L\big(P_{\max}-c_o\big)-c_m} \\
&\qquad +w\cdot c_m+\frac{c_o(-q_k^1+w\cdot c_m)}{P_{\max}-c_o}\Big) \\
\geq \; & m_1\beta_g+\sum_{k=1}^{m_1}\Big(\frac{(q_k^1+\beta_g)c_o}{\big(P_{\max}-c_o\big)} \\
&\qquad +w\cdot c_m+\frac{c_o(-q_k^1+w\cdot c_m)}{P_{\max}-c_o}\Big) \\
= \; & m_1\beta_g+\frac{m_1(\beta_g c_o+P_{\max}wc_m)}{P_{\max}-c_o} \\
= \; & \frac{m_1P_{\max}(\beta_g+wc_m)}{P_{\max}-c_o}.
\end{aligned}$$

□

Lemma 13.

$$\frac{\psi\left(1,p(\tau),e_i(\tau)\right)-c_m}{\psi\left(0,p(\tau),e_i(\tau)\right)-\psi\left(1,p(\tau),e_i(\tau)\right)+c_m} \geq \frac{c_o}{P_{\max}-c_o}.$$

Proof. We expand $\psi(y(\tau), p(\tau), e_i(\tau))$ for each case:

Case 1: $c_o \geq p(\tau)$. By Eqn. (2.9) and $e_i(\tau) \leq L, \forall i, \tau$,

$$\begin{aligned}\psi(1, p(\tau), e_i(\tau)) &= p(\tau)e_i(\tau) + c_m, \\ \psi(0, p(\tau), e_i(\tau)) &= p(\tau)e_i(\tau).\end{aligned}$$

Therefore,

$$\frac{\psi(1, p(t), e_i(t)) - c_m}{\psi(0, p(t), e_i(t)) - \psi(1, p(t), e_i(t)) + c_m} = \infty.$$

Case 2: $c_o < p(\tau)$. By Eqn. (2.9) and $e_i(\tau) \leq L, \forall i, \tau$,

Thus,

$$\begin{aligned}\psi(1, p(\tau), e_i(\tau)) &= c_o e_i(\tau) + c_m, \\ \psi(0, p(\tau), e_i(\tau)) &= p(\tau)e_i(\tau).\end{aligned}$$

Therefore,

$$\begin{aligned}&\frac{\psi(1, p(\tau), e_i(\tau)) - c_m}{\psi(0, p(\tau), e_i(\tau)) - \psi(1, p(\tau), e_i(\tau)) + c_m} \\ \geq\ &\frac{c_o e_i(\tau)}{p(\tau)e_i(\tau) - c_o e_i(\tau)} \\ \geq\ &\frac{c_o}{P_{\max} - c_o}.\end{aligned}$$

Combining both cases, we complete the proof of this lemma. □

□ **End of chapter.**

Appendix D

Proof of Theorem 9

Let $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ be an offline optimal solution obtained by solving **CP** and **EP** separately in sequence and $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ be an offline optimal solution obtained by solving the joint-optimization **DCM**. Let $\boldsymbol{x}^{\boldsymbol{on}}$ be the solution obtained by $\mathbf{GCSR}^{(\mathbf{w})}$ and $\boldsymbol{y}^{\boldsymbol{off}}$ be an offline optimal solution of **EP** given input $\boldsymbol{x}^{\boldsymbol{on}}$. Let $(\boldsymbol{x}^{\boldsymbol{on}}, \boldsymbol{y}^{\boldsymbol{on}})$ be the solution obtained by $\mathbf{DCMON}^{(\mathbf{w})}$. Denote $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}, \boldsymbol{y})$ to be cost of **DCM** of solution $(\boldsymbol{x}, \boldsymbol{y})$ and $\mathrm{C}_{\mathrm{CP}}(\boldsymbol{x})$ to be cost of **CP** of solution $\boldsymbol{x}$.

According to Theorem 8, equation (C.33) and the fact that the available look-ahead window size is only $[w - \Delta_s]^+$ for $\mathbf{DCMON}^{(\mathbf{w})}$ to solve **EP** (discussed in Sec. 5.3), we have

$$
\begin{aligned}
&\frac{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^{\boldsymbol{on}}, \boldsymbol{y}^{\boldsymbol{on}})}{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^{\boldsymbol{on}}, \boldsymbol{y}^{\boldsymbol{off}})} \\
\leq\ & 1 + \frac{2\beta_g \left(LP_{\max} - Lc_o - c_m\right)}{\beta_g LP_{\max} + [w - \Delta_s]^+ c_m P_{\max}\left(L - \frac{c_m}{P_{\max} - c_o}\right)} \\
\leq\ & 1 + \frac{2\left(LP_{\max} - Lc_o - c_m\right)}{LP_{\max} + \alpha_g P_{\max}\left(L - \frac{c_m}{P_{\max} - c_o}\right)} \\
\leq\ & 1 + 2\frac{P_{\max} - c_o}{P_{\max}} \cdot \frac{1}{1 + \alpha_g},
\end{aligned}
\tag{D.1}
$$

where $\triangle_s \triangleq \beta_s/(d_{\min} P_{\min})$ and $\alpha_g \triangleq \frac{c_m}{\beta_g}[w - \Delta_s]^+$ is a "normalized" look-ahead window size that takes values in $[0, +\infty)$.

According to Theorem 2, we have

$$
\frac{\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})}{\mathrm{C}_{\mathrm{DCM}}\left(\boldsymbol{x}^*, \boldsymbol{y}^*\right)} \leq \frac{LP_{\max}}{Lc_o + c_m}. \tag{D.2}
$$

Then if we can bound $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x}^{\boldsymbol{on}}, \boldsymbol{y}^{\boldsymbol{off}})/\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$, we obtain the competitive ratio upper bound of $\mathbf{DCMON}^{(\mathbf{w})}$. The following lemma gives us such a bound.

Lemma 14. $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x^{on}}, \boldsymbol{y^{off}})/\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) \leq 2 - \alpha_s$, *where* $\alpha_s \triangleq \min(1, w/\triangle_s)$ *and* $\triangle_s \triangleq \beta_s/(d_{\min} P_{\min})$.

Proof. It is straightforward that

$$\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x^{on}}, \boldsymbol{y^{off}}) \leq \mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x^{on}}, \bar{\boldsymbol{y}}). \tag{D.3}$$

So we seeks to bound $\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x^{on}}, \bar{\boldsymbol{y}})/\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$.

For solution $\boldsymbol{x^{on}}$ and $\bar{\boldsymbol{x}}$, denote

$$CW(\boldsymbol{x}) = \beta_s \sum_{t=1}^{T} [x(t) - x(t-1)]^+, \tag{D.4}$$

and

$$CI(\boldsymbol{x^{on}}, \bar{\boldsymbol{x}}) = \sum_{t=1}^{T} p(t) \left(d_t(x^{on}(t)) - d_t(\bar{x}(t))\right). \tag{D.5}$$

According to Eqn. (B.12), we have

$$CW(\boldsymbol{x_i^{on}}) = CW(\bar{\boldsymbol{x}}_i). \tag{D.6}$$

According to lemma 15 and the fact that $x_i^{on}(t), \bar{x}_i(t) \in \{0, 1\}, \ \forall t, i$, we have

$$\begin{aligned} CW(\boldsymbol{x^{on}}) &= CW(\sum_{i=1}^{M} \boldsymbol{x_i^{on}}) = \sum_{i=1}^{M} CW(\boldsymbol{x_i^{on}}) \\ &= \sum_{i=1}^{M} CW(\bar{\boldsymbol{x}}_i) = CW(\sum_{i=1}^{M} \bar{\boldsymbol{x}}_i) \\ &= CW(\bar{\boldsymbol{x}}), \end{aligned} \tag{D.7}$$

and

$$\begin{aligned} CI(\boldsymbol{x^{on}}, \bar{\boldsymbol{x}}) &= \sum_{t=1}^{T} p(t)\left(d_t(x^{on}(t)) - d_t(\bar{x}(t))\right) \\ &= \sum_{t=1}^{T} p(t)\left(\sum_{i=1}^{x^{on}(t)} d_t^i - \sum_{i=1}^{\bar{x}(t)} d_t^i\right) \\ &= \sum_{i=1}^{M}\sum_{t=1}^{T} p(t) d_t^i (x_i^{on}(t) - \bar{x}_i(t)) \\ &\leq (1-\alpha_s)\sum_{i=1}^{M} CW(\bar{\boldsymbol{x}}_{\boldsymbol{i}}) \\ &= (1-\alpha_s) CW(\bar{\boldsymbol{x}}), \end{aligned} \tag{D.8}$$

where the last and second last inequalities come from Eqns. (D.7) and (B.15), respectively.

According to Eqn. (2.9), we have $\forall b \in [0, x^{on}(t)]$,

$$\begin{aligned} & \psi\left(\bar{y}(t), p(t), d_t\left(x^{on}(t)\right)\right) - \psi\left(\bar{y}(t), p(t), d_t\left(b\right)\right) \\ \leq\; & p(t)\left(d_t\left(x^{on}(t)\right) - d_t\left(b\right)\right). \end{aligned} \tag{D.9}$$

By the definition of **DCM**, Eqns. (B.13), (D.4), (D.5) and (D.9),

$$\begin{aligned} & \mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x^{on}}, \bar{\boldsymbol{y}}) \\ = & \sum_{t=1}^{T} \{\psi\left(\bar{y}(t), p(t), d_t\left(x^{on}(t)\right)\right) \\ & +\beta_s[x^{on}(t) - x^{on}(t-1)]^+ + \beta_g[\bar{y}(t) - \bar{y}(t-1)]^+\} \\ \leq & \sum_{t=1}^{T} \{\psi\left(\bar{y}(t), p(t), d_t\left(\bar{x}(t)\right)\right) + p(t)\left(d_t\left(x^{on}(t)\right) - d_t\left(\bar{x}(t)\right)\right) \\ & +\beta_s[x^{on}(t) - x^{on}(t-1)]^+ + \beta_g[\bar{y}(t) - \bar{y}(t-1)]^+\} \\ = & \sum_{t=1}^{T} \left\{\psi\left(\bar{y}(t), p(t), d_t\left(\bar{x}(t)\right)\right) + \beta_g[\bar{y}(t) - \bar{y}(t-1)]^+\right\} \\ & +CW(\boldsymbol{x_{on}}) + CI(\boldsymbol{x_{on}}, \bar{\boldsymbol{x}}). \end{aligned} \tag{D.10}$$

Then, by Eqns. (D.7), (D.8) and (D.10), we have

$$\begin{aligned}
&\frac{\mathrm{C}_{\mathrm{DCM}}(\boldsymbol{x_{on}}, \bar{\boldsymbol{y}})}{\mathrm{C}_{\mathrm{DCM}}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})} \\
\leq\ & \frac{\sum_{t=1}^{T} \psi\left(\bar{y}(t), p(t), d_t\left(\bar{x}(t)\right)\right) + CI(\boldsymbol{x_{on}}, \bar{\boldsymbol{x}}) + CW(\boldsymbol{x_{on}})}{\sum_{t=1}^{T} \psi\left(\bar{y}(t), p(t), d_t\left(\bar{x}(t)\right)\right) + CW(\bar{\boldsymbol{x}})} \\
\leq\ & \frac{(1-\alpha_s)CW(\bar{\boldsymbol{x}}) + CW(\boldsymbol{x_{on}})}{CW(\bar{\boldsymbol{x}})} \\
=\ & \frac{(1-\alpha_s)CW(\bar{\boldsymbol{x}}) + CW(\bar{\boldsymbol{x}})}{CW(\bar{\boldsymbol{x}})} \\
=\ & 2-\alpha_s. \qquad \text{(D.11)}
\end{aligned}$$

This lemma follows from Eqns. (D.3) and (D.11). □

Theorem 9 follows from Eqns. (D.1), (D.2) and lemma 14.

Lemma 15. $\bar{\boldsymbol{x}}_{\boldsymbol{1}}, \bar{\boldsymbol{x}}_{\boldsymbol{2}}, \ldots \bar{\boldsymbol{x}}_{\boldsymbol{M}}$ *and* $\boldsymbol{x_1^{on}}, \boldsymbol{x_2^{on}}, \ldots \boldsymbol{x_M^{on}}$ *are decreasing sequences, i.e.,* $\forall t,\ \bar{x}_1(t) \geq ... \geq \bar{x}_M(t)$ *and* $x_1^{on}(t) \geq ... \geq x_M^{on}(t)$.

Proof. Recall that $\bar{\boldsymbol{x}}_{\boldsymbol{i}}$ and $\boldsymbol{x_i^{on}}$ are offline and online solutions obtained by $\mathbf{CPOFF_s}$ and $\mathbf{GCSR_s^{(w)}}$ for problem $\mathbf{CP}_{\mathrm{i}}$, respectively. According to the definition of $\mathbf{CP}_{\mathrm{i}}$, $a_1(t) \geq a_2(t) \geq ... \geq a_M(t)$ is a decreasing sequence and $d_t^1 \leq d_t^2 \leq ... \leq d_t^M$ is an increasing sequence. Thus, for problem $\mathbf{CP}_{\mathrm{i}}$, the larger the index i is, the more sparse workload tends to be and the higher power consumption tends to be. Hence, for a larger index i, there are more "idling intervals", meanwhile both $\mathbf{CPOFF_s}$ and $\mathbf{GCSR_s^{(w)}}$ tends to keep servers idling less during idling intervals (because idling cost is higher). So, $\bar{\boldsymbol{x}}_{\boldsymbol{1}}, \bar{\boldsymbol{x}}_{\boldsymbol{2}}, \ldots \bar{\boldsymbol{x}}_{\boldsymbol{M}}$ and $\boldsymbol{x_1^{on}}, \boldsymbol{x_2^{on}}, \ldots \boldsymbol{x_M^{on}}$ are decreasing sequences, i.e., $\forall t,\ \bar{x}_1(t) \geq ... \geq \bar{x}_M(t)$ and $x_1^{on}(t) \geq ... \geq x_M^{on}(t)$. □

□ **End of chapter.**

Bibliography

[1] Akamai tech. http://www.akamai.com.

[2] National solar radiation data base. http://rredc.nrel.gov/solar/old_data/nsrdb/1991-2010/.

[3] Nationalgrid. https://www.nationalgridus.com/.

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

[5] Solar panel. http://en.wikipedia.org/wiki/Solar_panel.

[6] Tecogen. http://www.tecogen.com.

[7] The weather channal. http://www.weather.com/.

[8] Apple's onsite renewable energy, 2012. http://www.apple.com/environment/renewable-energy/.

[9] Distributed generation, 2012. http://www.bloomenergy.com/fuel-cell/distributed-generation/.

[10] Energy independence, 2012. http://www.zdnet.com/want-energy-independence-5-on-site-generation-players-to-watch-7000002560/.

[11] C. Baldwin, K. Dale, and R. Dittrich. A study of the economic shutdown of generating units in daily dispatch. *IEEE Trans. Power Apparatus and Systems*, 1959.

[12] L. Barroso and U. Holzle. The case for energy-proportional computing. *IEEE Computer*, 2007.

[13] A. Beloglazov, R. Buyya, Y. Lee, and A. Zomaya. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in Computers*, 2011.

[14] A. Borbely and J. Kreider. *Distributed generation: the power paradigm for the new millennium*. CRC Press, 2001.

[15] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.

[16] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle. Managing energy and server resources in hosting centers. In *Proc. ACM SIGOPS*, 2001.

[17] A. Conejo, M. Plazas, R. Espinola, and A. Molina. Day-ahead electricity price forecasting using the wavelet transform and arima models. *IEEE Trans. Power Systems*, 2005.

[18] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1959.

[19] R. Doyle, J. Chase, O. Asad, W. Jin, and A. Vahdat. Model-based resource provisioning in a web service utility. In *Proc. USITS*, 2003.

[20] K. Fehrenbacher. ebay to build huge bloom energy fuel cell farm at data center. 2012. http://gigaom.com/cleantech/ebay-to-build-huge-bloom-energy-fuel-cell-farm-at-data-center/.

[21] K. Fehrenbacher. Is it time for more off-grid options for data centers?. 2012. http://gigaom.com/cleantech/is-it-time-for-more-off-grid-options-for-data-centers/.

[22] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper. Workload analysis and demand prediction of enterprise data center applications. In *IISWC*, 2007.

[23] S. Kazarlis, A. Bakirtzis, and V. Petridis. A genetic algorithm solution to the unit commitment problem. *IEEE Trans. Power Systems*, 1996.

[24] J. Koomey. Growth in data center electricity use 2005 to 2010. *Analytics Press*, 2010.

[25] W. Kwon and A. Pearson. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Trans. Automatic Control*, 1977.

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

[27] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser. Renewable and cooling aware workload management for sustainable data centers. In *Proc. ACM SIGMETRICS*, 2012.

[28] J. Lowesohn. Apple's main data center to go fully renewable this year. 2012. http://news.cnet.com/8301-13579_3-57436553-37/apples-main-data-center-to-go-fully-renewable-this-year/.

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

[30] V. Mathew, R. Sitaraman, and P. Shenoy. Energy-aware load balancing in content delivery networks. In *Proc. IEEE INFOCOM*, 2012.

[31] J. Muckstadt and R. Wilson. An application of mixed-integer programming duality to scheduling thermal generating systems. *IEEE Trans. Power Apparatus and Systems*, 1968.

[32] E. Nygren, R. Sitaraman, and J. Sun. The Akamai Network: A platform for high-performance Internet applications. 2010.

[33] N. Padhy. Unit commitment-a bibliographical survey. *IEEE Trans. Power Systems*, 2004.

[34] D. Palasamudram, R. Sitaraman, B. Urgaonkar, and R. Urgaonkar. Using batteries to reduce the power costs of internet-scale distributed networks. In *Proc. ACM Symposium on Cloud Computing*, 2012.

[35] S. Pelley, D. Meisner, T. Wenisch, and J. VanGilder. Understanding and abstracting total data center power. In *Workshop on Energy-Efficient Design*, 2009.

[36] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath. Load balancing and unbalancing for power and performance in cluster-based systems. In *Workshop on compilers and operating systems for low power*, 2001.

[37] T. Shiina and J. Birge. Stochastic unit commitment problem. *International Trans. Operational Research*, 2004.

[38] M. Stadler, H. Aki, R. Lai, C. Marnay, and A. Siddiqui. Distributed energy resources on-site optimization for commercial buildings with electric and thermal storage technologies. *Lawrence Berkeley National Laboratory*, 2008.