Online Microgrid Energy Generation Scheduling Revisited: The Benefits of Randomization and Interval Prediction

Mohammad H. Hajiesmaili[†], Chi-Kin Chau[‡], Minghua Chen[†], and Longbu Huang[§]

[†]Department of Information Engineering, The Chinese University of Hong Kong
[‡]Masdar Institute of Science and Technology, UAE
[§]Institute for Interdisciplinary Information Sciences, Tsinghua University
{mohammad, minghua}@ie.cuhk.edu.hk, ckchau@masdar.ac.ae, longbohuang@tsinghua.edu.cn

ABSTRACT

Energy generation scheduling is a fundamental problem in microgrid design that determines the on/off status and the output level of energy sources with the goal of minimizing the cost and satisfying both electricity and heat demand. The uncertainty in both renewable generation and microgrid demand makes the problem drastically different from its counterparts and in traditional power systems and brings out the essential need of online algorithm design. In the literature, an online deterministic algorithm called CHASE has achieved a competitive ratio of 3, which is the best possible among deterministic algorithms. In addition, it has been shown the accurate prediction can improve the performance. This paper revisits the problem by investigating the benefits of randomization and interval prediction, *i.e.*, relaxing accurate prediction assumption by considering an interval of valid ranges for future demand. We propose *r*CHASE, a randomized algorithm that achieves competitive ratio of around 2.128, improving beyond the best deterministic algorithm. Then, we propose *i*CHASE, an interval prediction-aware algorithm that is built upon *r*CHASE and a new extension we developed for the classic ski-rental problem. Our trace-driven experiments demonstrate that *i*CHASE outperforms CHASE; the average cost reduction of *i*CHASE is 15.85%, while CHASE reduces the cost by 9.1%.

CCS Concepts

•Hardware → Smart grid; •Theory of computation → *Online algorithms;* Scheduling algorithms;

Keywords

Microgrids; energy generation scheduling; randomized online algorithm; interval prediction

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

e-Energy'16, June 21-24, 2016, Waterloo, ON, Canada

© 2016 ACM. ISBN 978-1-4503-4393-0/16/06…$15.00

DOI: http://dx.doi.org/10.1145/2934328.2934329

1. INTRODUCTION

Nowadays, microgrid is regarded as a promising paradigm of future power systems. In microgrid, various energy sources such as renewable energy sources (*e.g.*, wind farms), local generation units (*e.g.*, gas generators), together with the external grid contribute in fulfilling the time-varying energy demand (electricity and heat) of a campus-scale local community. As compared to traditional grids, microgrid has recognized advantages in cost efficiency, environmental awareness, and power reliability. Consequently, worldwide installed microgrid capacity has witnessed a rapid growth, reaching $1283MW$ in 2015, and is expected to reach $2855MW$ by 2020 in the US only [6].

In microgrid, intelligent energy generation scheduling is a key mechanism that can reduce the operational microgrid cost significantly [19]. The scheduling policy aims (i) to schedule the output level of local generators and on-demand request from the external grid and natural gas provider to satisfy the net demand, and (ii) to minimize the operational cost which is consisted of the generator's cost and the cost of acquiring energy from external grid and natural gas provider. This problem is appeared previously in traditional power system literature as economic dispatching [11] or unit commitment [17] problems. In microgrid, however, the problem is more involved mainly because of the unprecedented uncertainty in net demand that is originated from unpredictability, uncontrollability, and intermittency of both renewable sources and microgrid demand [19].

In the previous study [18], the problem has been tackled following a competitive online algorithm design approach. Toward this, an online deterministic algorithm called CHASE has been proposed based on a special structure of offline optimal solution. Then, it is shown that CHASE achieves *optimal* competitive ratio of 3, *i.e.*, no other deterministic algorithm can achieve a competitive ratio of better than 3. In addition, CHASE is extended to leverage the limited prediction of the accurate future data, such as near-term demand or wind forecast. As a result, it is shown that accurate prediction of near-future demand, can further improve the performance of the basic prediction-oblivious CHASE. In practice, however, it is extremely difficult to *accurately* predict the future net demand given (i) the significant uncertainties involved in the output of the renewable sources, (ii) abrupt changes in small-scale microgrid demand without periodical aggregation effect, and finally (iii) additional dimension of uncertainty due to co-generation, since electricity and heat demands exhibit rather different stochastic

Reference	General approach	Start-up cost	Future information	Randomized Design	Co-generation	Peak charging
Narayanaswamy *et al.* [19]	OCO	✗	✓	✗	✗	✗
Zhang *et al.* [30]	CAD	✗	✗	✓	✗	✓
Lu *et al.* [18]	CAD	✓	✓(accurate)	✗	✓	✗
This work	CAD	✓	✓(interval prediction)	✓	✓	✗

Table 1: Summary and comparison of the previous work and this work; CAD: Competitive algorithm design; OCO: Online convex optimization

patterns that impose further difficulty to accurately predict the overall combined demand.

This paper revisits the online energy generation scheduling problem in microgrids by investigating the theoretical and practical benefits of the following two design space issues in algorithm design:

1. **Randomized online algorithm design.** Randomization, *i.e.*, making random algorithmic decisions based on some probability distribution functions, is a well-established technique in online algorithm design [26] that in several cases can result in better performance guarantee in theory and/or in practice.

2. **Interval prediction-aware online algorithm design.** A promising alternative approach against accurate prediction of future demand is to leverage *interval predictions*, which is extensively studied in forecasting wind power generations and residential loads in microgrids [22, 25, 27, 29]. In this approach, the idea is to predict an interval for the possible values for each future input. In other word, in this model, an *upper bound* and a *lower bound* of the future demand are predicted. While there are a plenty of studies on leveraging the benefits of future accurate prediction in algorithm design, incorporating interval prediction in algorithm design is not extensively studied in the past.

Summary of contributions. By investigating the benefits of randomization and interval prediction in online algorithm design for microgrid energy generation scheduling problem, this paper makes the following contributions:

- We propose *r*CHASE, a randomized algorithm that like deterministic CHASE tries to *chase* the optimal offline solution. Different from CHASE, it makes on/off decisions randomly based on two special distribution functions. We prove *r*CHASE achieves competitive ratio of around 2.128, improving the best deterministic algorithm CHASE.

- We propose *i*CHASE, an online algorithm that leverages interval prediction in design. The algorithm is built upon *r*CHASE and a new algorithm design for basic ski-rental problem taking into account the interval prediction of skiing days. We show that the competitive ratio of *i*CHASE could not be worse than that of *r*CHASE. Furthermore, as compared to the existing modified CHASE with accurate prediction, *i*CHASE is more realistic, since it requires the interval information of the future demand instead of accurate point forecast. Last but not the least, our interval prediction design provides a general approach towards solving similar online problems with "*buy-or-rent*" nature.

- By trace-driven experiments, we evaluate the performance of our proposed algorithms. The average results spanned over a whole year demonstrate that the cost of *i*CHASE with interval prediction with $\pm 40\%$ deviation from accurate future values achieves the operational cost of $1.12\times$ and $1.08\times$ of offline-optimal and CHASE with perfect future prediction, respectively. The results also show that *i*CHASE outperforms CHASE; the average cost reduction of *i*CHASE is 15.85%, while basic CHASE reduces the cost by 9.1%.

In the rest of the paper, Sec. 2 reviews the related work. Sec. 3 formulates the problem. Sec. 4 reviews the existing algorithm CHASE. *r*CHASE and *i*CHASE are proposed in Secs. 5 and 6, respectively. Experiments are reported in Sec. 7, and finally, Sec. 8 concludes the paper.

2. RELATED WORK

Energy generation scheduling in microgrids. Providing cost-effective energy generation scheduling mechanism for "grid-connected" microgrids has been studied extensively in the recent studies. The most related problems to ours are those that try to propose *online* solutions for energy generation scheduling problem in microgrid [18, 19, 30]. There are several aspects both in problem formulation and solution approaches that make the works to be different. We compare the most important aspects of these works and our work in Table 1. The authors in [19] apply online convex optimization framework [31] to design economic dispatching algorithms for microgrids. In [30], the economic dispatching problem with peak-based charging model in competitive online design framework is studied. Both of the above works focus on the economic dispatching problem that does not take into account the start-up cost. It is worth noting that incorporating start-up cost and turning the problem to joint unit commitment and economic dispatch make the problem fundamentally different.

The problem studied in this paper is a joint economic dispatching and unit commitment problem with co-generation taken into account that is proposed in [18] for the first time in microgrids. This paper improves the obtained results in [18] by (i) obtaining better bounds on theoretical performance guarantee of [18] by utilizing randomized algorithm design, and (ii) making the algorithms more practical by leveraging the interval prediction in algorithmic design.

Energy generation scheduling in traditional power systems. Energy generation scheduling is a well-studied

problem in traditional power systems. Two main variants are unit commitment [12, 17, 20] and economic dispatching [10,11,21]. The unit commitment problem typically optimizes the on/off (start-up and shut-down) schedule of power generation units in large power systems and it is known to be a challenging NP-complete problem in general [12]. In the previous studies, different approaches have been used to approximately solve this problem. Some approaches are mix-integer programming [9], dynamic programming [23], and stochastic programming [24]. Economic dispatching problem, however, takes the output of unit commitment problem (on/off status of power generation units) as its input and tries to schedule the units by determining their output levels. There are different works toward solving economic dispatching problem in large scale traditional power systems [10, 21]. There is also interest in involving CHP generators in economic dispatching problem to satisfy both electricity and heat demand simultaneously [13].

All of the aforementioned studies try to propose offline solutions, which is reasonable in traditional large-scale power systems with high aggregation effect on demand and low penetration of unpredictable renewable sources. These two issues results in ability to accurately predict the whole time horizon demand and then solve the problem in offline in which the demand and energy supply (or at least their distributions) in the entire time horizon are known *a prior*. In microgrid, offline solutions are not applicable, because small-scale demand and renewable output are uncontrollable, unpredictable, and intermittence [28]. There are a few studies for *offline* energy generation scheduling in microgrid scenarios as well. We refer to [15, 16] as samples.

3. ENERGY GENERATION SCHEDULING PROBLEM

The system model and problem formulation are based on that of [18]. Consider a microgrid in which local generation units along with renewable sources are solicited to fulfill both electricity and heat demands, increase the reliability, and reduce total energy cost. Hence, the problem of interest could be casted as a microgrid cost minimization problem (MCMP) with demand (both electricity and heat) covering constraints and generator's capacity and physical constraints. By summarizing the main notations in Table 2, we proceed to explain the system model.

3.1 System Model

First, we assume that the system runs in a time-slotted manner in a time horizon of T time slots in which the length of each time slot is fixed in order of one hour.

Energy demand: The energy demand is two-fold. (i) Net electricity demand denoted by $a(t)$ at time slot t, which is the total electricity demand subtracted by the output of the renewable sources. (ii) Heat demand denoted by $h(t)$ at time t. To capture any desired level of uncertainty, we do not assume any specific stochastic model of $a(t)$ and $h(t)$.

External grid and heat: Our model assumes that the microgrid works in the "connected-mode", *i.e.,* in addition to covering the demand by local generation and renewable sources, the residual demand is fulfilled by acquiring the electricity from the main grid in an on-demand manner. Following the widely adapted time-of-use pricing policy [4], let the spot grid price at time t be $p(t)$ where

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

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

$p(t) \in [P_{\min}, P_{\max}]$. Again, there is no assumptions on the underlying stochastic model of $p(t)$. Finally, heating can be generated separately using external natural gas, which costs c_g per unit of demand.

Local generation: There is a local generator with capacity L.[1] Following the commonly-adapted generator model [17], let β as the startup cost of turning on a generator, c_m as the sunk cost of maintaining a generator in its active state per unit time, and c_o as the operational cost per unit time for an active generator to output one unit of energy. In realistic generators there are minimum turning on/off periods as well as ramping up/down constraints, which imposes some additional constraints to the main problem. In [18], a general approach towards tackling the cases with this set of constraints is proposed, thereby, in this paper we focus on the cases in which the generators are so-called "fast-responding" ones, *i.e.*, the only constraint is the capacity limit of each generation unit. Finally, local generators can generate electricity and heat simultaneously. Let the heat recovery efficiency for co-generation be η, *i.e.,* for each unit of electricity generated, η unit of useful heat can be supplied for free. Thus,

[1]Our proposed algorithms in this paper can be extended to the case with multiple homogeneous generators, without additional performance loss. The general approach is to decompose the problem by following a layering approach which is proposed in [18, Sec. 3.3].

ηc_g is the saving due to using co-generation to supply heat, provided that there is sufficient heat demand. We assume $c_o \geq \eta \cdot c_g$, *i.e.*, it is cheaper to acquire heat by natural gas than purely by generators.

3.2 Problem Definition

Optimization variables: Let $v(t)$ and $s(t)$ be the amount of electricity and heat obtained by the external grid and the external natural gas, respectively. For the generator, let $y(t)$ be the binary on/off status (1 as on and 0 as off) and $u(t)$ as the output level.

Objective function: The objective function is in the form of aggregated operational cost over time horizon $\mathcal{T}$ that is formulated as follows:

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

where $\psi\big(\sigma(t),y(t)\big) \triangleq p(t)v(t) + c_g s(t) + c_o u(t) + c_m y(t)$, in which the first two terms are the grid and external gas costs, and the second two terms are the operating cost of local generator. This operational cost $\psi\big(\sigma(t),y(t)\big)$ along with the generator's switching cost $\beta[y(t)-y(t-1)]^+$ is aggregated over the entire $\mathcal{T}$. In addition, we assume generator is off in the beginning, *i.e.,* $y(0)=0$.

Microgrid Cost Minimization Problem (MCMP): Given the above optimization variables and objective function, the optimization problem is formulated as follows:

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

where the packing constraint (1b) ensures respecting the capacity limit of the generator. Covering constraints (1c)-(1d) force to fulfill the electricity and heat demands by grid, natural gas, and the generator.

Remark that MCMP is challenging to solve since: (i) the problem is a mixed-integer linear problem, which is generally difficult to tackle even in offline, where complete knowledge of future information is available in advance; (ii) the objective function is coupled over the time because of the startup cost $\beta[y(t)-y(t-1)]^+$ term in the objective, thereby MCMP cannot be horizontally decomposed over time; (iii) in practice, the problem data, *i.e.,* the demands $a(t)$ and $h(t)$ as well as the grid spot price $p(t)$ arrive online, which emphasizes the need for online solution design.

4. OVERVIEW OF THE EXISTING SOLUTIONS

In this section, we briefly overview the existing solution to MCMP that is proposed in [18]. The structure of the optimal offline solution and the previous online algorithm are used in the following sections to propose new randomized and interval prediction-aware algorithms in this paper.

4.1 Optimal Offline Algorithm Design

The first observation is that when the on/off status is given, MCMP degenerates into a timely decoupled linear program, whose optimal solution has a closed-form structure as stated in Lemma 1.

LEMMA 1. *[18] Given a fixed on/off status $\big(y(t)\big)_{t=1}^T$, the solution that minimizes $\mathsf{cost}(y,u,v,s)$ is*

$$u(t)=\begin{cases} 0, & \text{if} \quad p(t)+\eta\cdot c_g \leq c_o,\\ \min\Big\{\frac{h(t)}{\eta}, a(t), Ly(t)\Big\}, & \text{if} \quad p(t)<c_o<p(t)+\eta\cdot c_g,\\ \min\Big\{a(t), Ly(t)\Big\}, & \text{if} \quad c_o\leq p(t), \end{cases} \tag{2}$$

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

Applying the result in Lemma 1, MCMP can be further simplified to problem sMCMP with respect to a single variable y as follows:

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

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

Now, we proceed to overview the offline optimal solution (in offline setting we assume that the input $\big[\sigma(t)\big]_{t=1}^T$ is known a priori) to sMCMP. This offline solution is leveraged to design the our randomized online algorithm in Sec. 5. Define

$$\delta(t) \triangleq \psi(\sigma(t),0) - \psi(\sigma(t),1), \tag{4}$$

as the single-slot cost difference between using or not using local generation, without taking into account the start-up cost. When $\delta(t)>0$ (resp. $\delta(t)<0$), the generator must be turned on (resp. off). The startup cost, however, prevent us to turn on/off frequently. Hence, the question is on how to make on/off decision with start-up cost? The idea is to evaluate whether the *cumulative* gain/loss in the future can offset the startup cost or not. To realize this idea the *cumulative cost difference* $\Delta(t)$ is defined as

$$\Delta(t) \triangleq \min\Big\{0, \max\{-\beta, \Delta(t-1)+\delta(t)\}\Big\}, \tag{5}$$

where the initial values is $\Delta(0)=-\beta$ and we have $\Delta(t)\in[-\beta,0]$, by definition.

Having defined function Δ, the next step is to build *critical segments*, which is the key in offline algorithm design. Toward this, we partition the time horizon $\mathcal{T}$ into several disjoint sets called *critical segments*. By introducing a set of *critical slots* $T_1^c < T_2^c < ... < T_k^c$, the critical segments are defined as $[1,T_1^c],[T_1^c+1,T_2^c],[T_2^c+1,T_3^c],\ldots,[T_k^c+1,T]$. Furthermore, each critical slot T_i^c comes with an auxiliary slot $\tilde{T}_i^c$, such that the pair $(T_i^c,\tilde{T}_i^c)$ satisfies the following conditions:

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

Each pair of $(T_i^c,\tilde{T}_i^c)$ corresponds to an interval where $\Delta(t)$ traverses from -β to 0 or from 0 to -β, while in the middle never touches the boundaries. In [18], it is shown that such segments are well-defined and unique. An example of such partitioning into critical segments is shown in Fig. 1.

The next step is to categorize the critical segments based on their boundary values. By setting $T_0^c=0$ and $T_{k+1}^c=T$, we define the following categories of the critical segments:

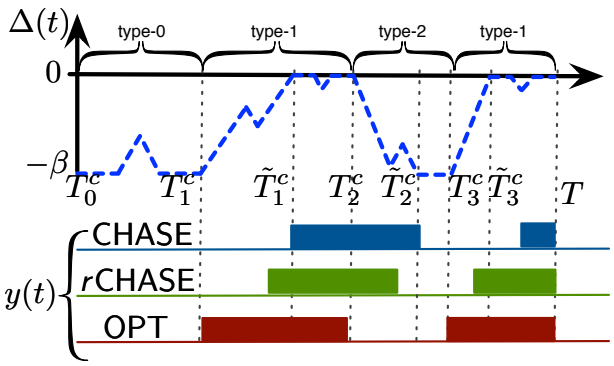


Figure 1: An example of $\Delta(t)$ and behaviors of CHASE [18], rCHASE (proposed in this paper), and OPT (that calculates the optimal offline solution)

- **type-**0: $[1, T_1^c]$
- **type-**1: $[T_i^c+1, T_{i+1}^c]$, if $\Delta(T_i^c) = -\beta$ and $\Delta(T_{i+1}^c) = 0$
- **type-**2: $[T_i^c+1, T_{i+1}^c]$, if $\Delta(T_i^c) = 0$ and $\Delta(T_{i+1}^c) = -\beta$
- **type-**3: $[T_k^c + 1, T]$

By this critical segment definition, the important result in [18] is mentioned in the following theorem.

THEOREM 1. *[18] An optimal solution of sMCMP is*

$$y^\star(t) \triangleq \begin{cases} 1, & \text{if } t \in \textbf{type-1}, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

Putting together the results in Theorem 1 and Lemma 1, it is straightforward to compute the optimal offline solution of sMCMP. On top of that, these results provide the key insight in tackling the problem in online setting.

4.2 Online Algorithm Design

In this section, we explain the online algorithm CHASE that is proposed in [18]. Recall that in optimal offline solution, we set $y(t) = 1$ right after entering **type-**1 segments, and $y(t) = 0$ by entering to the other segments. In online setting without future information, however, the challenge originates from the fact that it is impossible to immediately detect segment type right after entering it.

The good news is that even though immediate segment type detection is impossible, it is possible to detect the segment type eventually. This is done for **type-**1, right after $\Delta(t)$ reaches 0 for the first time after hitting $-\beta$. The similar procedure can be done for the other types. This observation is the key in proposing CHASE in [18] (summarized as Algorithm 1). In CHASE, $y(t) = y(t-1)$ if $-\beta < \Delta(t) < 0$, since changing segment type is not possible. However, when $\Delta(t) = 0$ (resp. $\Delta(t) = -\beta$), it is ensured that the segment is **type-**1 (resp. **type-**2). Thereby, CHASE sets $y(t) = 1$ (resp. $y(t) = 0$). In this way, CHASE tries to to track the offline optimal in an online manner since the status is changed right after ensuring that the segment type is changed.

Despite its apparent simplicity, the following results proves that CHASE is a competitive algorithm with the best possible competitive ratio among all the possible deterministic online algorithms for problem sMCMP.

Algorithm 1: [18] CHASE, for $t \in \mathcal{T}$

```
find Δ(t)
if Δ(t) = −β then
    y(t) ← 0
else if Δ(t) = 0 then
    y(t) ← 1
else
    y(t) ← y(t − 1)
end
set u(t), v(t), and s(t) according to Eqs. (2) and (3)
```

THEOREM 2. *[18] CHASE is $(3-2\alpha)$-competitive algorithm, where $0 < \alpha \leq 1$ is a parameter that depends on the problem input. Moreover, no other online algorithm can achieve better competitive ratio than $(3-2\alpha)$.*

Finally, we remark that CHASE is a single-generator version of the algorithm. Extending the algorithm for multiple generator case follows a carefully designed layering approach that preserves the competitiveness of single-generator CHASE [18].

In the next section, we propose a new randomized algorithm called *r*CHASE and explain how randomized decision making can play its role in improving the competitive ratio of the deterministic CHASE.

5. RANDOMIZED ONLINE ALGORITHM

In several theoretical online combinatorial problems, it turns out that randomized algorithms, *i.e.*, those that make random decisions based on some probability distribution functions, might outperform the deterministic ones. This approach is called *randomized approximation algorithm design* and the expected value of the randomized solution is analyzed in performance (competitive) analysis [26].

Designing randomized algorithms, however, is highly involved in problem's structure. Overall, the high level idea is to follow the deterministic algorithm, but, make the decisions in randomized manner. In other words, while deterministic decision making turns the design to be too *conservative*, randomized algorithm tries to be more *aggressive* by randomized decision making, and consequently improve the performance and in practice and in theory by achieving good *expected* competitive ratio.

As stated in the previous section, the idea behind the deterministic CHASE algorithm is to track the optimal offline solution in online fashion. However, CHASE is too conservative in making decisions: it waits until to be ensured that the offline optimal solution has switched $y(t)$, then it follows the offline optimal. The main idea behind the randomized algorithm proposed in this section is that it might be better to be more aggressive in making decisions. Consequently, the goal in randomized algorithm is somehow switch $y(t)$ before $\Delta(t)$ reaches either 0 or $-\beta$.

Proper design of randomized algorithm is not straightforward, because of the following two key challenges:

1. how to make randomized decisions such that keep the algorithm to be reasonably aggressive, *i.e.,* to prevent it from frequent status change of the generator; recall that each status change imposes a potential start-up cost β to the overall cost.

2. how to generate random variables such that the expected performance guarantee (competitive ratio) could be obtained. Essentially, to address this challenge, designing a proper probability distribution function for random decision making is required.

Algorithm 2: *r*CHASE for $t \in \mathcal{T}$

find $\Delta(t)$
if $\Delta(t) = 0$ **then**
  generate γ_{off} according to $f_{\mathsf{off}}(\gamma)$ as in Eq. (8)
end
if $\Delta(t) = -\beta$ **then**
  generate γ_{on} according to $f_{\mathsf{on}}(\gamma)$ as in Eq. (7)
end
if $\Delta(t) \geq \gamma_{on}$ **then**
  $y(t) \leftarrow 1$
  $\gamma_{\mathsf{on}} \leftarrow 0$
else if $\Delta(t) \leq \gamma_{off}$ **then**
  $y(t) \leftarrow 0$
  $\gamma_{\mathsf{off}} \leftarrow -\beta$
else
  $y(t) \leftarrow y(t-1)$
end
set $u(t)$, $v(t)$, and $s(t)$ according to Eqs. (2) and (3)

We address aforementioned two challenges by designing a randomized algorithm called *r*CHASE that is summarized as Algorithm 2. In what follows we explain the detail of the algorithm and its behavior as compared to the existing deterministic CHASE.

The main idea behind *r*CHASE is as follows. Whenever, $\Delta(t)$ reaches 0 (resp. $-\beta$), a continuous random variable γ_{off} (resp. γ_{on}) is generated in $[-\beta, 0]$ with probability distribution function $f_{\mathsf{off}}(\gamma)$ (resp. $f_{\mathsf{on}}(\gamma)$) that is defined in Eq. (8) (resp. Eq. (7)). At time slot t, if $\Delta(t)$ is greater than or equal to random variable γ_{on}, the generator is turned on, (*i.e.,* $y(t) \leftarrow 1$). In addition, it sets γ_{on} to be 0, *i.e.,* no random decision making on turning on anymore. This is the key to prevent the algorithm to be too aggressive in changing the status. In other words, this remedies challenge (1). Similar scenario is appeared for turning off the generator: at time slot t, if $\Delta(t)$ is smaller than or equal to random variable γ_{off}, the generator is turned off, (*i.e.,* $y(t) \leftarrow 0$). In addition, it sets γ_{off} to be β to prevent random decision making in turning off. Finally, to address challenge (2), we design the following probability distribution functions:

$$f_{\mathsf{on}}(\gamma) = \begin{cases} \frac{C_1}{2\beta+\gamma} + C_2\delta(\gamma), & \text{if } -\beta < \gamma \leq 0, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

and

$$f_{\mathsf{off}}(\gamma) = \begin{cases} \frac{C_1}{\beta-\gamma} + C_2\delta(\gamma+\beta), & \text{if } -\beta \leq \gamma < 0, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where $C_1 = 2/(4\ln 2 - 1)$ and $C_2 = (2\ln 2 - 1)/(4\ln 2 - 1)$ are two constants, and $\delta(\gamma)$ is Dirac Delta function[2]. Note that *r*CHASE reduces to CHASE by setting $\gamma_{\mathsf{on}} = 0$ and $\gamma_{\mathsf{off}} = -\beta$.

An example of *r*CHASE as compared to CHASE is shown in Fig. 1. As compared to deterministic decision made by CHASE, random variable generation in *r*CHASE, *i.e.,* randomly generating γ_{off} and γ_{on} between $[-\beta, 0]$, causes earlier generator switch which makes it to be closer to the optimal offline decision making. However, it is worth noting that *r*CHASE is randomized, and in some intervals it may change the generator status too early or change the status whenever optimal offline does not change. Our theoretical result in Theorem 3, demonstrates that the expected competitive ratio of *r*CHASE is better as compared to the performance of CHASE.

THEOREM 3. *rCHASE for problem MCMP has a competitive ratio of* $1 + \frac{2}{4\ln 2 - 1} \approx 2.128$.

PROOF. In [14]. □

Remarks. The result in Theorem 2 [18] reports the competitive ratio of $3 - 2\alpha$ for CHASE in which α is a constant that depends on the system parameters, and in worst case α approaches 0, when the generator unit cost is much lower than the maximum grid spot price. Hence competitive ratio of CHASE approaches 3. Furthermore in [18], it is shown that no other deterministic algorithm can achieve better competitive ratio than that of CHASE. Result in Theorem 3 says that *r*CHASE is a 2.128-competitive algorithm which evinces 29% improvement as compared to the competitive ratio of CHASE. In addition, we note that even though the general structure of both algorithms looks similar, the techniques that are used in the analysis of *r*CHASE are fundamentally different from those in CHASE. We remark that the proposed algorithm *r*CHASE is for the single-generator scenario. Extension to the multiple generator case requires following a particular layering approach that is proposed in [18, Sec. 3.3]. Finally, we emphasize that the proposed algorithm includes fast-responding generators. One approach to extend the algorithm to the slow-responding generators is to project the obtained output levels of *r*CHASE to the values that respect the ramping constraints of the generator. For details we refer to [18, Sec. 4].

6. ONLINE ALGORITHM DESIGN WITH INTERVAL PREDICTION

In this section, we design another algorithm called *i*CHASE with interval prediction taking into account. First, we begin by defining the concept of interval prediction.

6.1 The Definition of Interval Prediction

It is known that accurate prediction of near-future demand, can further improve the performance of the basic prediction-oblivious CHASE [18]. In practice, however, it is extremely difficult to accurately predict the future net demand given (i) the significant uncertainties involved in the output of the renewable sources and (ii) unpredictably in small-scale microgrid demand without aggregation effect.

A promising alternative approach is to advocate the idea of interval prediction instead of accurate point forecasting, *i.e.*, predicting a range instead of exact value prediction in which prediction errors could be usually unavoidable [22,25, 27,29]. In this approach, the idea is to predict an interval for

[2]Dirac delta function, is a function on the real number line that is zero everywhere except at zero, with an integral of one over the entire real line.

the demand of each time slot. In other word, in this model, an *upper bound* and a *lower bound* of the future demand are predicted. Putting together, one can say, an interval is predicted for the value of the demand for each slot.

For our algorithm design in this section, we assume that the interval prediction of the future demand is available in a time window of size ω slots in addition to the current and the previous demand profile. In particular, we assume that the upper bound demand profile $\big(\overline{\sigma}(\tau)\big)_{\tau=t+1}^{t+\omega}$ and the lower bound demand profile $\big(\underline{\sigma}(\tau)\big)_{\tau=t+1}^{t+\omega}$ are given as additional inputs to the algorithm, such that the actual demand profile are within these values, *i.e.,* $\underline{\sigma}(\tau) \leq \sigma(\tau) \leq \overline{\sigma}(\tau), \forall \tau \in [t+1, t+\omega]$. In practice, there are several analytical and experimental approaches to calculate these values. Some examples are [7] for short-term load forecast of microgrids and [27] for interval prediction of wind sources.

Our idea in this section is based on a simple, yet effective extension to incorporate the interval prediction in algorithm design. In Sec. 6.2, we first explain the high level idea on how to extend the basic online algorithm for the well-known classic ski-rental problem [8] by taking into account the interval prediction, and discuss how this idea can improve the performance of the online algorithm in both theory and practice. Then in Sec. 6.3, by leveraging this idea, we propose our algorithm *i*CHASE that is built upon randomized *r*CHASE and our new extension to ski-rental problem.

6.2 Motivational Example on Ski-rental Problem

The ski-rental problem is as follows. Assume that renting skis costs 1 per time slot and buying skis costs B. The player has to decide in an online fashion whether to continue renting ski or buy skis in order to minimize the cost. The adversary, on the other hand, selects the total duration of skiing x to maximize the cost. In this way, the offline optimal solution would be $\min\{x, B\}$. In the literature it is known that the online algorithm for this problem is to rent the skis for the first B days, and then buy the skis. This algorithm achieves the competitive ratio of 2 which is the best possible among all online deterministic algorithms.

Now, we incorporate interval prediction in the basic setting of ski-rental problem. We consider a generalized problem, such that the player at any time slot t has a prediction of range of the total duration of skiing $[\underline{x}(t), \overline{x}(t)]$, such that x must fall in the range $[\underline{x}(t), \overline{x}(t)]$. In the basic setting of the classic ski-rental problem, the range is $[0, \infty]$.

We consider the deterministic online algorithm $\mathcal{A}$ that is summarized as Algorithm 3. In a nutshell of algorithm $\mathcal{A}$, the effectiveness of the additional interval data (*i.e.,* $\overline{x}(t)$ and $\underline{x}(t)$) is examined first. If at least one of the two additional data is useful, *i.e.,* either $\overline{x}(t)$ or $\underline{x}(t)$ can assure better worst-case performance than the basic online algorithm, 2-competitiveness, we change the deterministic decision making. Otherwise, we follow the basic algorithm. The following theorem characterizes the competitiveness of algorithm $\mathcal{A}$.

THEOREM 4. *The competitive ratio of* $\mathcal{A}$ *is* $\alpha \triangleq \min\Big\{2, \max\{\frac{B}{\underline{x}}, 1\}, \max\{\frac{\overline{x}}{B}, 1\}\Big\}$. *Note that* $1 \leq \alpha \leq 2$.

PROOF. First, we note that the competitive ratio of break-even is 2 [8].

Algorithm 3: $\mathcal{A}$ for $t \in \mathcal{T}$

```
1 if B/x(t) ≥ 2 AND x̄(t)/B ≥ 2 then
2 |   rent skis until the end or B, then buy skis
3 else if B/x(t) ≤ x̄(t)/B then
4 |   buy skis from the beginning
5 else
6 |   rent skis until the end
7 end
```

Second, if the player buys ski from at time t based on Line 4 of algorithm $\mathcal{A}$, then the competitive ratio is

$$\frac{B+t}{\min\{x, B\}} \leq \frac{B+t}{\min\{\underline{x}, B\}} \leq \max\{\frac{B}{\underline{x}}, \frac{t}{B}\} \leq \max\{\frac{B}{\underline{x}}, 1\}.$$

Third, if the player rents skis until the end, then the competitive ratio is

$$\frac{x}{\min\{x, B\}} \leq \max\{\frac{x}{B}, 1\} \leq \max\{\frac{\bar{x}}{B}, 1\}.$$

The proof is completed, when we consider the option that induces the least competitive ratio. □

6.3 Algorithm Design with Interval Prediction for MCMP

Now, based on the simple idea introduced in the previous section, we proceed to design an algorithm that incorporates interval prediction in decision making for online microgrid energy generation scheduling problem.

In the case of MCMP, depending the current status of the generator, the goal is to answer one of the following four questions:

1. If the generator is off, how to use the interval prediction data to keep the generator to be off?
2. If the generator is off, how to use the interval prediction data to turn on the generator?
3. If the generator is on, how to use the interval prediction data to turn off the generator?
4. If the generator is on, how to use the interval prediction data to keep the generator to be on?

In what follows, we answer to these questions by providing a guideline for each one. First, we start with some notation definition to facilitate the discussion. Let t_s be the index of the latest slot in which the status of the generator is changed due to the execution of the algorithm, *i.e.*,

$$t_s \triangleq \max\Big\{\tau = 1, \ldots, t-1 | y(\tau) \neq y(\tau - 1)\Big\}.$$

Then, we calculate the total cost of the previous time slots in $[t_s, t-1]$ denoted as Ψ as follows

$$\Psi(t) \triangleq \sum_{\tau=t_s}^{t-1} \psi(\sigma(\tau), y(\tau)). \tag{9}$$

Note that $y(\tau) = c, \tau \in [t_s, t-1], c \in \{0, 1\}$, by definition. Finally, define parameter th which is the competitive ratio of *r*CHASE. This parameter th is used to verify the effectiveness of interval prediction data. In Sec. 6.3.1, we proceed to

Algorithm 4: *i*CHASE, for $t \in \mathcal{T}$

1 **inputs**
2 $y(t-1)$ // The previous status of the generator
3 $(\overline{\sigma}(\tau))_{\tau=t+1}^{t+\omega}$ // The upper bounds of input in $[t+1, t+\omega]$
4 $(\underline{\sigma}(\tau))_{\tau=t+1}^{t+\omega}$ // The lower bounds of input in $[t+1, t+\omega]$
5 **algorithm**
6 $t_s \leftarrow \max\left\{\tau = 1, \ldots, t-1 | y(\tau) \neq y(\tau-1)\right\}$
7 $\Psi(t) \leftarrow \sum_{\tau=t_s}^{t-1} \psi(\sigma(\tau), y(t-1))$
8 **if** $y(t-1) = 0$ **then**
9 $\quad \overline{R}_0(t) \leftarrow \min_{t \le \tau' \le t+\omega} \frac{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\overline{\sigma}(\tau), 0)}{\Psi(t) + \beta + \sum_{\tau=t}^{\tau'} \psi(\underline{\sigma}(\tau), 1)}$
10 $\quad \underline{R}_0(t) \leftarrow \min_{t \le \tau' \le t+\omega} \max\left\{1 + \frac{\Psi(t)}{\beta}, \frac{\Psi(t) + \beta + \sum_{\tau=t}^{\tau'} \psi(\overline{\sigma}(\tau), 1)}{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\underline{\sigma}(\tau), 0)}\right\}$
11 $\quad$ **if** $\overline{R}_0(t) \ge$ *th AND* $\underline{R}_0(t) \ge$ *th* **then**
12 $\quad\quad$ execute *r*CHASE
13 $\quad$ **else if** $\underline{R}_0(t) \le \overline{R}_0(t)$ **then**
14 $\quad\quad y(t) \leftarrow 1$
15 $\quad$ **else**
16 $\quad\quad y(t) \leftarrow 0$
17 $\quad$ **end**
18 **else**
19 $\quad \overline{R}_1(t) \leftarrow \min_{t \le \tau' \le t+\omega} \frac{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\overline{\sigma}(\tau), 1)}{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\underline{\sigma}(\tau), 0)}$
20 $\quad \underline{R}_1(t) \leftarrow \max_{t \le \tau' \le t+\omega} \frac{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\overline{\sigma}(\tau), 0)}{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\underline{\sigma}(\tau), 1)}$
21 $\quad$ **if** $\overline{R}_1(t) \ge$ *th AND* $\underline{R}_1(t) \ge$ *th* **then**
22 $\quad\quad$ execute *r*CHASE
23 $\quad$ **else if** $\underline{R}_1(t) \le \overline{R}_1(t)$ **then**
24 $\quad\quad y(t) \leftarrow 0$
25 $\quad$ **else**
26 $\quad\quad y(t) \leftarrow 1$
27 $\quad$ **end**
28 **end**
29 set $u(t)$, $v(t)$, and $s(t)$ according to Eqs. (2) and (3)

answer questions (1) and (2) in which the current status of the generator is off. Then in Sec. 6.3.2, we explain how to answer questions (3) and (4) which are related to the case that the generator is currently on.

6.3.1 Guidelines when the generator is off

In this case, we have $y(t-1) = 0$ and the decision is whether turn on the generator or not. Overall goal is to make the decision based on comparing the values of the following two ratios:

$$\overline{R}_0(t) \triangleq \max_{t \le \tau' \le t+\omega} \frac{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\overline{\sigma}(\tau), 0)}{\Psi(t) + \beta + \sum_{\tau=t}^{\tau'} \psi(\underline{\sigma}(\tau), 1)}, \quad (10)$$

$$\underline{R}_0(t) \triangleq \min_{t \le \tau' \le t+\omega} \frac{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\overline{\sigma}(\tau), 1) + \beta}{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\underline{\sigma}(\tau), 0)}. \quad (11)$$

In Eq. (10), $\overline{R}_0(t)$ characterizes the maximum ratio between the maximum possible cost of using the grid over the minimum possible cost using the generator. Similarly, in Eq. (11), $\underline{R}_0(t)$ characterizes the minimum ratio between the maximum possible cost of using generator over the minimum cost of using the grid.

Our decision making is as follows. If both values of $\overline{R}_0(t)$ and $\underline{R}_0(t)$ are greater than the threshold th, the additional information on upper bound and lower bound cannot provide any useful insight. Then, we follow the basic algorithm without interval prediction taking into account, like *r*CHAESE (Line 12 of Algorithm 4). This is similar to what we have done in Line 2 of algorithm $\mathcal{A}$.

The interval prediction information actively contributes in decision making when either $\overline{R}_0(t) \le$ th or $\underline{R}_0(t) \le$ th. If so, we compare those values and make the decision based on the smaller one. If $\underline{R}_0(t) \le \overline{R}_0(t)$, it means that we can safely turn on the generator without imposing significant additional cost (Line 14, this is like buying decision in Line 4 of algorithm $\mathcal{A}$), otherwise, we keep using the gird (Line 16, this is like renting decision in Line 6 of algorithm $\mathcal{A}$). By this design we answer questions (1) and (2).

6.3.2 Guidelines when the generator is on

In this case, we have $y(t-1) = 1$ and the decision is whether keep the generator on or turn it off and switch to use the external grid. Similar to the previous case, we construct the following two ratios which are later used for decision making:

$$\overline{R}_1(t) \triangleq \min_{t \le \tau' \le t+\omega} \frac{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\overline{\sigma}(\tau), 1)}{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\underline{\sigma}(\tau), 0)}, \quad (12)$$

$$\underline{R}_1(t) \triangleq \max_{t \le \tau' \le t+\omega} \frac{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\overline{\sigma}(\tau), 0)}{\Psi(t) + \sum_{\tau=t}^{\tau'} \psi(\underline{\sigma}(\tau), 1)}. \quad (13)$$

In Eq. (12), $\overline{R}_1(t)$ characterizes the minimum ratio between the maximum cost of keep using the generator over the minimum cost occurred by using the grid. Likewise, in Eq. (13), $\underline{R}_1(t)$ characterizes the maximum possible ratio between the maximum cost of switching to the grid over the minimum cost of when keep turning on the generator.

Similar to the previous case we first verify whether the values of $\overline{R}_1(t)$ and $\underline{R}_1(t)$. If both are greater than the threshold th, then we simply execute *r*CHAESE (Line 22 of Algorithm 4). Otherwise, if $\overline{R}_1(t)$ is smaller, we keep using the generator (Line 26) and if $\underline{R}_1(t)$ is smaller, we turn off the generator (Line 24). This answers questions (3) and (4).

Putting together the above guidelines, we summarize *i*CHASE as in Algorithm 4. The following theorem establishes the competitiveness of *i*CHASE.

THEOREM 5. *The competitive ratio of iCHASE is at most the competitive ratio of rCHASE.*

PROOF. In [14]. □

Acronym	Description
OPT	Optimal offline solution
CHASE	The basic algorithm in [18] without lookahead information (Algorithm 1)
CHASE10	The algorithm in [18] with lookahead window size $\omega = 10$
*i*CHASE	Randomized algorithm with interval prediction (Algorithm 4)

Table 3: Acronyms for the algorithms

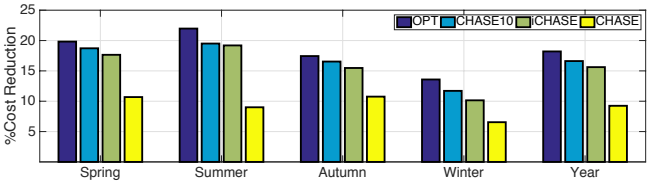


Figure 2: Comparison results in different seasons

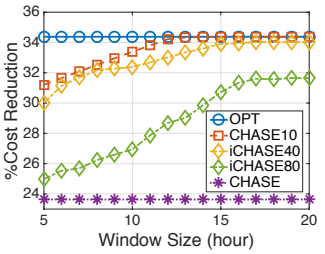


Figure 3: Cost reduction as a function of look ahead window size

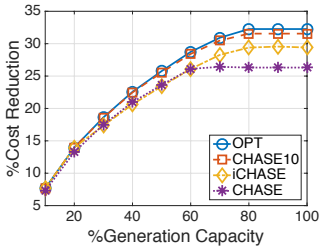


Figure 4: Cost reduction as a function of local generation capacity

7. PERFORMANCE EVALUATIONS

In this section, we conduct trace-driven experiments to evaluate the performance of the proposed algorithms and compare them to the previous algorithms. As the performance metric, in our experiments we report the cost reduction of different algorithms as compared to the cost incurred by using only external electricity, heating and wind energy (without the generators) as a benchmark. Furthermore, the output for all algorithms is the total cost incurred during the time horizon $\mathcal{T}$. Finally, the length of each time slot is 1 hour and $T = 120$ which is an interval of 5 days.

7.1 Experimental Setup

Overview of data traces: The electricity and heat data are obtained from [1] which is the demand of a college in San Francisco with yearly electricity demand of around $154GWh$. To calculate the net demand, the renewable energy supply is injected by a wind power trace from [3] which is the output of a wind station outside San Francisco with installed capacity of $12MW$. The electricity and natural gas price data are from PG&E [4]. The electricity prices follow a time-of-use scheme in which there are three different prices for on-peak, off-peak, and mid-peak periods. According to [2], we set the unit heat generation cost to be $c_g = \$0.0179/KWh$. We adopt generators with specifications the same as the one in [5] with the capacity of $L = 3MW$, The incremental cost per unit time $c_o = \$0.051/KWh$ and $c_m = \$110/h$ which is calculated according to the natural gas price and the generator efficiency. The heat to electricity ratio η is set to be 1.8 according to [5]. We set the startup cost β equivalent to running the generator at its full capacity for about 5 hours at its own operating cost, which gives $\beta = \$1400$. Even though we explained the algorithms in the single-generator case, but, as mentioned in Sec. 3.1, our algorithms can be extended to the multiple-generator case without performance loss. Hence, in our experiments, we run our algorithms with 10 homogeneous generators so as to fully satisfy the demand.

Comparison algorithms: As summarized in Table 3, in our experiments we report the cost reduction of the following algorithms:

1) OPT: The optimal offline solution which calculates the optimal energy generation scheduling provided that the problem data over the whole time horizon $\mathcal{T}$ is available in advance.

2) CHASE: The basic algorithm proposed in [18] that is also explained in Sec. 4 as Algorithm 1.

3) CHASE10: The lookahead extended version of CHASE, with window of size 10. In this algorithm that is proposed in [18], it is assumed that accurate data of the future 10 time slots are available at the current slot.

4) *i*CHASE: The randomized algorithm that incorporates the interval prediction in algorithm design as explained in Sec. 6.3 as Algorithm 4. We remark that since *i*CHASE is built upon *r*CHASE, we just report the results for *i*CHASE directly.

The settings of iCHASE algorithm: Unless otherwise mentioned, we assume that in a window size of 10 time slots, the interval prediction demand data is available. We generate this interval prediction data randomly such that the upper bound and lower bound values are within $\pm 40\%$ of the original points, on average. In other words, at time t we have

$$\overline{\sigma}(\tau) = \big(1 + \mathsf{rnd}(0, 0.8)\big)\sigma(\tau), \tau \in [t+1, t+\omega] \qquad (14)$$

$$\underline{\sigma}(\tau) = \big(1 - \mathsf{rnd}(0, 0.8)\big)\sigma(\tau), \tau \in [t+1, t+\omega] \qquad (15)$$

where $\mathsf{rnd}(a, b)$ uniformly generates a random number in $[a, b]$. Since we have assumed that the upper bound and lower bound values are within $\pm 40\%$ of the original points, *on average*, we have generated the random values within $\big(1 - \mathsf{rnd}(0, 0.8)\big)$ and $\big(1 + \mathsf{rnd}(0, 0.8)\big)$ of the original inputs. Two sample interval predicted values along with the accurate values for the net electricity demand (*i.e.,* $a(t)$) are demonstrated in Fig. 5. Finally, note that *i*CHASE calls the randomized algorithm *r*CHASE in some slots. In addition, the upper and lower bound values are generated randomly in our experiments. Consequently, we have executed *i*CHASE 200 times in each data point of the figures for the same input and the average cost reduction values are reported.

7.2 Comparison Results in Different Seasons

In the first experiment we compare the cost reduction of 4 different algorithms in 4 seasons and whole year. Toward this for each season, we have executed 15 runs for each experiment in a period of 5 days with coverage of 75 days in each season, and the average cost reduction results for different

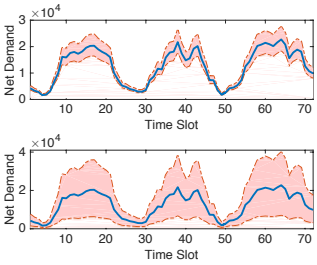


Figure 5: A sample illustration of net electricity demand and the interval prediction with 20% (top) and 80% (bottom) of the values

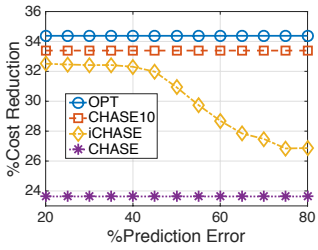


Figure 6: Cost reduction as a function of interval prediction size

algorithms separately for each season are depicted in Fig. 2. We mention the following notable observations: (i) The average cost reduction values for whole year demonstrate that *i*CHASE reduces the benchmark cost by 15.85% as compared to 16.61% reduction for CHASE10. This observation shows that *i*CHASE can reduce the overall cost close to CHASE10. Note that CHASE10 requires accurate demand profile within lookahead window of 10 hours, which is not practical in real cases with high uncertainty in electricity demand, heat demand, and renewable generations. While iCHASE just requires an estimation on upper and lower bound values within $\pm 40\%$ of the accurate points. (ii) The cost reduction of *i*CHASE (15.85%) is significantly better than that of the basic prediction-oblivious CHASE (9.1%). (iii) Last but not the least, *i*CHASE's cost is $1.12\times$ of the optimal offline cost, which evinces the importance of randomization and taking into account interval prediction in algorithm design.

7.3 The Effect of Lookahead Window

In this section, we aim to investigate the evolution of the cost reduction of *i*CHASE as compared to CHASE10, as the size of lookahead window increases. Toward this, we change the window size from 5 to 20. For the sake of better illustration, we report the results of two versions of *i*CHASE, named as *i*CHASE40 and *i*CHASE80 with interval prediction within $\pm 40\%$ and $\pm 80\%$ of the original points, on average respectively. In Fig. 3, the cost reduction of a sample run as a function of lookahead window size is demonstrated. The results demonstrate that by sufficiently large window size ($\omega \approx 15$), *i*CHASE40 (with 33.95% reduction) can achieve $1.01\times$ of the optimal offline cost (with 34.37% reduction) and $1.007\times$ of the cost of perfect prediction CHASE10 (with 34.21% reduction). In addition, even a low quality interval prediction like *i*CHASE80 (with 30.71% reduction at $\omega = 15$) can perform much better than the basic CHASE without any future information (with 23.61% reduction).

7.4 The Effect of Generation Capacity

Another interesting scenario is to investigate the performance of the proposed algorithms when the generation capacity increases. As an important observation in [18], it is reported that "provisioning local generation to produce 60% of the peak demand is sufficient to obtain nearly all of the cost reduction benefits". In this experiment, we aim to verify this observation again taking into account the new design spaces of randomization and interval prediction in our algorithms. Toward this, we change the number of generators from 1 (corresponding to 10% of the peak demand) to 10 (corresponding to 100% of the peak demand) and report the cost reduction results of different algorithms in Fig. 4. The notable result is that the cost reduction of our algorithm *i*CHASE, evolves the same as the other alternatives, which signifies again the aforementioned observation on the capacity of local generation.

7.5 The Effect of the Accuracy of Interval Prediction

In the last experiment, we investigate the effect of the accuracy of the interval prediction on the amount of cost reduction of *i*CHASE. Toward this, we vary the interval prediction bounds from $\pm 20\%$ to $\pm 80\%$ according to the Eqs. (14) and (15). Note that for the previous experiments interval prediction is within $\pm 40\%$ of the original values, on average. An example of such interval predictions for the boundary points $\pm 20\%$ to $\pm 80\%$ for 3 days is depicted in Fig. 5. The result of cost reduction of *i*CHASE as the accuracy of prediction decreases is shown in Fig. 6. The result clearly demonstrates that the *i*CHASE is robust to the error from $\pm 20\%$ to $\pm 40\%$ and when the prediction error goes over 40%, the performance degrades rapidly as the accuracy decreases.

8. CONCLUSION AND FUTURE WORK

This paper investigated the potential benefits of randomization and interval prediction in online algorithm design for intelligent energy generation scheduling in microgrid. Our analytical study demonstrated that randomization can improve the competitiveness of existing deterministic algorithms. Our experimental study demonstrated that new design space of randomization and interval prediction can significantly improve the result of the previous deterministic algorithm and can achieve near offline-optimal performance.

An important future work is to find the best randomized competitive ratio for the studied problem. In addition, it would be interesting to extend the idea of interval prediction-aware algorithmic design in peak-based charging model. Finally, another interesting future direction would be to consider more realistic prediction models such as the case that the prediction accuracy deceases along the increasing of the predicted time horizon.

Acknowledgment

The authors would like to thank Jinlong Tu for the contributions on in the initial stage of the work. The work presented in this paper was supported in part by National Basic Research Program of China (Project No. 2013CB336700) and the University Grants Committee of the Hong Kong Special Administrative Region, China (Theme-based Research Scheme Project No. T23-407/13-N and General Research Fund No. 14201014). The work of Longbo Huang was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61361136003, 61303195, Tsinghua Initiative Research Grant, Microsoft Research Asia Collaborative Research Award, and the China Youth 1000-talent Grant.

9. REFERENCES

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

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

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

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

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

[6] http://www.greentechmedia.com/.

[7] N. Amjady, F. Keynia, and H. Zareipour. Short-term load forecast of microgrids by a new bilevel prediction strategy. *Smart Grid, IEEE Transactions on*, 1(3):286–294, Dec 2010.

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

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

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

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

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

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

[14] M. H. Hajiesmaili, C.-K. Chau, M. Chen, and L. Huang. Online microgrid energy generation scheduling revisited: The benefits of randomization and interval prediction. *Technical report, https://staff.ie.cuhk.edu.hk/ mhchen/papers/CHASE-R.tr.pdf*.

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

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

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

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

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

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

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

[22] N. A. Shrivastava, K. Lohia, and B. K. Panigrahi. A multiobjective framework for wind speed prediction interval forecasts. *Renewable Energy*, 87:903–910, 2016.

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

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

[25] D. Torregrossa, J.-Y. Le Boudec, and M. Paolone. Model-free computation of ultra-short-term prediction intervals of solar irradiance. *Solar Energy*, 124:57–67, 2016.

[26] V. V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[27] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong. Optimal prediction intervals of wind power generation. *Power Systems, IEEE Transactions on*, 29(3):1166–1174, 2014.

[28] J. Wang, M. Shahidehpour, and Z. Li. Security-constrained unit commitment with volatile wind power generation. *IEEE Trans. Power Systems*, 23(3):1319–1327, 2008.

[29] G. Zhang, Y. Wu, K. P. Wong, Z. Xu, Z. Y. Dong, and H.-C. Iu. An advanced approach for construction of optimal wind power prediction intervals. *Power Systems, IEEE Transactions on*, 30(5):2706–2715, Sept 2015.

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

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