Generalized Exact Scheduling: a Minimal-Variance Distributed Deadline Scheduler

Yorie Nakahira

Carnegie Mellon University, yorie@cmu.edu

Andres Ferragut

Universidad ORT Uruguay, ferragut@ort.edu.uy

Adam Wierman

California Institute of Technology, adamw@caltech.edu

Many modern schedulers can dynamically adjust their service capacity to match the incoming workload. At the same time, however, unpredictability and instability in service capacity often incur operational and infrastructural costs. In this paper, we seek to characterize optimal distributed algorithms that maximize the predictability, stability, or both when scheduling jobs with deadlines. Specifically, we show that *Exact Scheduling* minimizes both the stationary mean and variance of the service capacity subject to strict demand and deadline requirements. For more general settings, we characterize the minimal-variance distributed policies with soft demand requirements, soft deadline requirements, or both. The performance of the optimal distributed policies is compared to that of the optimal centralized policy by deriving closed-form bounds and by testing centralized and distributed algorithms using real data from the Caltech electrical vehicle charging facility and many pieces of synthetic data from different arrival distributions. Moreover, we derive the Pareto-optimality condition for distributed policies that balance the variance and mean square of the service capacity. Finally, we discuss a scalable partially-centralized algorithm that uses centralized information to boost performance and a method to deal with missing information on service requirements.

Key words: Deadline scheduling, Service capacity control, Exact Scheduling, Online distributed algorithm

1. Introduction

Traditionally, the scheduling literature has assumed a static or fixed service capacity. However, it is increasingly common for modern applications to have the ability to dynamically adjust their service capacity in order to match the current demand. For example, power distribution networks match the energy supply to changing demand. When using cloud computing services, one can modify the total computing capacity by changing the number of computing instances and their speeds. The ability to adapt service capacity dynamically gives rise to challenging new design questions. In particular, how to enhance the predictability and stability of service capacity is of great importance in such applications since peaks and fluctuations often come with significant costs [1–3]. For example, in the emerging load from electric vehicle charging stations, maintaining stable power consumption (*i.e.* limiting the fluctuations in power consumption) is important because large peaks in power use may strain the grid infrastructure and result in a high peak charge for the station operators. The stations also prefer predictable power consumption (*i.e.* knowing future power consumption) because purchasing power in real time is typically more expensive than purchasing in advance. Cloud content providers also prefer stable and predictable service capacity because on-demand contracts for compute instances (*e.g.* Amazon EC2 and Microsoft Azure) are typically more expensive than long-term contracts. Additionally, significant fluctuations in service capacity induce unnecessary power consumption and infrastructure strain for computing equipment.

1

Thus, in situations where service capacity can be dynamically adjusted, an important design goal is to reduce the costs associated with unpredictability and instability in the service capacity (Figure 2) while maintaining a high quality of service, *e.g.* meeting job deadlines and satisfying job demands. In this work, we study this problem by designing policies that minimize the variance of the service capacity in systems where jobs arrive with demand and deadline requests. Our model is motivated by power distribution networks, where the size of jobs and (active) service capacity are small compared to the total energy resources available and where contracts often depend on the mean and variance of service capacity, *e.g.* if a charging station participates in the regulation market, then costs/payments rely explicitly on them [4, 5].

Although the literature on deadline scheduling is large and varied, optimal algorithms are only known for certain niche cases. We review some of these results in the related work section below. We emphasize however that only recently have researchers approached the task of designing algorithms that balance service quality and costs associated with variability. Much of the work on this topic has been application-driven, particularly in the areas of cloud computing and power distribution systems. As we mentioned before, in these areas service capacity is indeed elastic by design, and variability has direct cost implications. In this regard, no general optimality results have been proven so far about how to balance service quality and cost, except in some limited settings, such as deterministic worst-case settings [6], single server systems [7–9], and/or heavy traffic settings [10, 11]. In heavy traffic settings, the dynamic behavior can be approximated by a continuous-state process involving Brownian motion, for which there exist established tools to optimize. On the other hand, optimizing queueing systems without continuous-state approximations remains a hard problem. Solving this problem is a challenging task due to the heterogeneity of jobs (diversity in service requests) and the size of the state and decision space (numbers of possible configurations on existing job profiles and the set of feasible control policies).

In this paper, we consider the cases when the system needs to schedule jobs to achieve high quality of service while maintaining predictability and stability of the service capacity. We seek to design optimal *distributed algorithms*, which only use local information about each job to decide the desirable service rate. We consider cases with stationary and non-stationary arrival distributions under strict or soft service constraints. We derive general optimality results that hold beyond the heavy-traffic regime. The algorithms we study are particularly useful for large systems such as power distribution networks and cloud computing, where implementing centralized algorithms is likely to be prohibitively slow and costly in large-scale service systems, *i.e.* we are unlikely to be able to access global information about all jobs and servers in real time when deciding the service rate of individual jobs. Despite such constraints on information sharing, we show that, interestingly, the optimal distributed algorithms under mild assumptions have comparable performance to centralized algorithms.

Contributions of this paper. In this paper, we adapt tools from optimization and control theory to characterize the optimal distributed policies in a broad range of settings without any approximation (see Table 1 for a summary of the results). Further, we provide a novel performance bound that describes the gap between the performance of optimal distributed policies and the performance of optimal centralized policies.

Specifically, we identify the optimal distributed algorithms under strict demand and deadline requirements (Theorem 1), soft demand requirements (Theorem 2), soft deadline requirements (Theorem 3), and soft demand and deadline requirements (Theorem 4) in settings with stationary Poisson arrivals as well as non-stationary Poisson arrivals (Theorem 5 and Corollary 3).

Our first results focus on stationary arrivals. While a considerable amount of work has analyzed the variance of specific policies (see [12] and references therein), little prior work characterizes the optimal policies. In the basic setting of strict service requirements, we show that *Exact Scheduling* is the optimal distributed algorithm, *i.e.* the distributed algorithm that minimizes the stationary service capacity variance. Exact Scheduling is a simple scalable algorithm that works by finishing jobs *exactly* at their deadlines using a constant service rate [3, 12, 13]. Although it has received

(a) Predictability + penalties for unsatisfied service

	Stationary arrivals	Non-stationary arrivals
Problem formulation	Section 3.1	Section 5.1
Strict demands and deadlines	Section 3.2, Theorem 1	Section 5.2, Theorem 5
Soft demands and strict deadlines	Section 3.3, Theorem 2	Section 5.2, Corollary 3
Strict demands and soft deadlines	Section 3.4, Theorem 3	Section 5.2, Corollary 3
Soft demands and strict deadlines	Section 3.5, Theorem 4	Section 5.2, Corollary 3

(b) Predictability + stability

	Non-stationary arrivals
Problem formulation	Section 5.1
Predictability	Section 5.2, Theorem 5
Stability	Section 5.3, Corollary 4
Balancing predictability and stability	Section 5.4, Theorem 6

Table 1   Summary of the main results

considerable attention in the existing literature, its optimality conditions have been unknown. In more general settings of soft service requirements, we propose novel generalizations of Exact Scheduling, each of which minimizes a combination of the service capacity variance, the expected penalties for unsatisfied demands, and the expected penalties for deadline extensions. These optimal algorithms all have closed-form expressions and use constant service rates with varying forms of rate and admission control. Due to these properties, they are also easy to implement and highly scalable.

We also extend our results to the case of non-stationary Poisson job arrivals and characterize the pareto-optimal algorithm that balances service capacity variance, penalties for unsatisfied demands, and penalties for deadline extensions. Additionally, we consider a more general class of objective functions: the service capacity variance, the mean-squared service capacity, and the weighted sum of the two. The resulting optimal algorithm has a striking analogy to the YDS algorithm [14], which is an offline algorithm that minimizes service capacity peaks in a related, deterministic worst-case version of the problem. This connection suggests the opportunity to transform other deterministic offline algorithms to stochastic online algorithms in related problems.

Given our focus on *distributed* algorithms, an important question is how these distributed algorithms perform compared with the optimal centralized algorithm. However, a major difficulty comes from the fact that the optimal centralized algorithms are unknown and no bounds on the optimal cost exist. Leveraging tools from optimal control, we provide closed-form formulas on the performance degradation due to using distributed algorithms (Lemma 3 and Corollary 2). The resulting bounds suggest that, when sojourn times are homogeneous, Exact Scheduling attains the optimal trade-off between service capacity variance and total remaining demand variance achievable by any centralized algorithms. Note that our proof technique (Lemma 3) is novel in its use of optimal control and has the potential for providing performance bounds for other scheduling policies. We also compare distributed algorithms with centralized algorithms in our motivating examples of electric vehicle charging. Our test in Caltech electric vehicle charging testbed [15] shows that the proposed optimal distributed algorithms also achieve comparable performance with existing centralized algorithms in practice.

Related work. There is extensive literature that studies the design and analysis of deadline scheduling algorithms, particularly in the computing literature (*e.g.* [3,16] and references therein). The most classical deadline scheduling algorithms include Earliest Deadline First (EDF) [7–9,17] and Least Laxity First (LLF) [17], among others [18,19]. These algorithms are originally developed for situations when jobs must be completed before their deadlines, and their analysis primarily focused on the rate or amount of job completion as opposed to the service capacity predictability.

The EDF policy is shown to maximize the number of completed jobs in the single-server case when reneging occurs after service [8]. On the other hand, the LLF policy is optimal when reneging occurs before entering service. The LLF policy is also known to maximize the amount of completed work in a multiserver system that models EV charging [20].

In the context of manufacturing systems, optimal schedules were derived for systems where tasks are flexible in time and variable service (machine production) speed [21]. The problem of minimizing the total completion time of a sequence of tasks was studied for manufacturing systems with variable service capacity [22]. In this context, the sequence of tasks is fixed and given beforehand, and optimization can be performed offline, something that we explore in our papers when considering offline optimal algorithms. However, in our work we are concerned with online distributed schedules and consider instead a steady arrival of new tasks. Instead of focusing on total completion time, we explore a tradeoff between users perceived performance and resource predictability.

In modern applications such as cloud computing and smart grid, it is important to study the trade-offs between service quality and costs associated with variability in service capacity [12,23,24]. In the context of cloud computing, algorithms have been proposed to control the variability of power usage in data centers using deferrable jobs (see [25–30] and references therein). In the context of power distribution systems, algorithms have been designed to control the variability of energy supply using deferrable loads (see [31–34] and references therein). The focus of this line of work is to track a given power profile and provide regulation service to the network (*i.e.* reducing variability and ensuring predictability). The most common approach to do so is the use of model predictive control (MPC), which solves optimization problems at each time and may not be scalable.

The optimal schedulers for deadline scheduling have been obtained for deterministic worst-case settings [6,14], single server systems [7–9], and/or heavy traffic [10,11]. In the heavy traffic analysis, the dynamic behavior of discrete queueing systems is often approximated by a continuous-state process of stochastic differential equations.

2. Model description

The goal of this paper is to characterize the online scheduling policies that minimize service capacity variance, mean square, and both subject to service quality constraints for systems with the ability to dynamically adjust their service capacity. We use a continuous time model, where $t \in \mathcal{T} = [0, T]$ denotes a point in time and $T \geq 0$ is the (potentially infinite) time horizon. Each job, indexed by $k \in \mathcal{V} = \{1, 2, \cdots\}$, is characterized by an arrival time a_k, a service demand σ_k, a sojourn time τ_k, a unit cost for unsatisfied demand δ_k, and a unit cost for deadline extension ϵ_k. Given the arrival time a_k and the sojourn time τ_k, the deadline of job k is defined to be $a_k + \tau_k$. Before we formulate the scheduler design problem, we first introduce below the arrival profiles, the service profiles, and the design objectives.

Arrival profiles. We represent the set of jobs as a marked point process $\{(a_k; \sigma_k, \tau_k, \delta_k, \epsilon_k)\}_{k \in \mathcal{V}}$ in space $\mathcal{T} \times S \times C$, where the arrival times $a_k \in \mathcal{T}$ are the set of points, and the service requirements $(\sigma_k, \tau_k) \in S$ and costs for unmet requirements $(\delta_k, \epsilon_k) \in C$ are the set of marks.[1] We assume that the point process is an independently marked Poisson point process, which is defined by an intensity function $\tilde{\Lambda}(a)$ on $\mathcal{T}$ and a mark joint density measure $f_a(\sigma, \tau)g_a(\delta)h_a(\epsilon)$ on $S \times C$ [35]. This also implies that $\{(a_k; \sigma_k, \tau_k)\}_{k \in \mathcal{V}}$ is a Poisson point process on $\mathcal{T} \times S$ with the intensity function $\Lambda(a, \sigma, \tau) = \tilde{\Lambda}(a)f_a(\sigma, \tau)$. We assume that $f_a(\sigma, \tau)$ has a bounded support. Intuitively, the intensity function is the average rate at which jobs with service requirement (σ, τ) arrive at time a. When both $\tilde{\Lambda}(a) \equiv \Lambda$ and $f_a(\sigma, \tau) \equiv f(\sigma, \tau)$ do not depend on a, we say that the arrival distribution is stationary . For a stationary arrival distribution, the intensity function of the Poisson point process simplifies to $\Lambda f(\sigma, \tau)$. We focus on stationary arrival processes in Section 3 and then generalize our results to non-stationary arrivals in Section 5. Throughout, we assume that the service demand σ

---

[1] Here, we use $(a; \sigma, \tau, \delta, \epsilon)$ to denote the random variables and $(a_k; \sigma_k, \tau_k.\delta_k, \epsilon_k)$ to denote one realization of them in job k.

and the sojourn time τ have finite first and second moments, S is bounded, and $S \subset \{(\sigma, \tau) : \tau \geq \sigma$ and $\sigma \geq 0\}$.[2]

Service profiles. The service system works on each job $k \in \mathcal{V}$ with a *service rate* $r_k(t)$, which is an integrable function of t. The service rate can take any non-negative values that are smaller than the maximum rate $\bar{r}_k$, *i.e.*

$$r_k(t) \in [0, \bar{r}_k], \tag{1}$$

and without loss of generality, we assume that $\bar{r}_k = 1$. To meet the demand requirements, the service rate must satisfy

$$\int_{a_k}^{\infty} r_k(t)dt = \sigma_k, \qquad\qquad \forall k \in \mathcal{V}. \tag{2}$$

To meet the deadline requirements, it needs to satisfy

$$\begin{aligned} r_k(t) &= 0, \quad \forall a_k < t, \forall k \in \mathcal{V}, \\ r_k(t) &= 0, \quad \forall t > a_k + \tau_k, \forall k \in \mathcal{V}, \end{aligned} \tag{3}$$

where $\mathbf{1}\{\epsilon\}$ denotes the indicator function for an event ϵ.

The service rate also determines three important quantities associated with costs: service capacity, the amount of unsatisfied demands, and the amount of deadline extensions. The service capacity is the instantaneous resource consumption of the system, given by

$$P(t) = \sum_{k \in \mathcal{V}} r_k(t).$$

We assume that $P(t)$ has no upper bound, implying that there is always enough capacity to serve the jobs. The total penalty for unmet demands of jobs with deadline t is

$$U(t) = \sum_{k \in \mathcal{V}: a_k + \tau_k = t} \delta_k (\sigma_k - \hat{\sigma}_k),$$

where $\hat{\sigma}_k = \int_{a_k}^{\infty} r_k(t)dt$ is defined to be the satisfied demands of job k. The total penalty for deadline extensions of jobs with deadline t is

$$W(t) = \sum_{k \in \mathcal{V}: a_k + \tau_k = t} \epsilon_k (\hat{\tau}_k - \tau_k).$$

where $\hat{\tau}_k = \min\{\max\{t - a_k : r_k(t) > 0\}, T\}$ is defined to be the actual sojourn time of job k.

Design objectives. We consider designing *online* scheduling algorithms, which decide the service rates in real-time without using the future job arrival information. For scalability, we restrict our attention to *distributed* algorithms which only need local information about each job to decide its service rate. Examples of online distributed algorithms are *Immediate Scheduling, Delayed Schedule,* and *Exact Scheduling* (see Figure 1).

Recall from Section 1 that the predictability and stability of service capacity are important design criteria for modern schedulers because peaks and fluctuations in service capacity strain the system infrastructure and knowing the future demand of the service capacity helps reduce cost (Figure 2). Thus, our design objective is to reduce the variance and mean square in service capacity

---

[2] The condition $\tau \geq \sigma$ constrains the service demand σ of a job to be no more than the amount of service that can be provided within its sojourn time τ.

for the settings with strict or soft service constraints. Specifically, we consider the optimization problem

$$\text{minimize} \quad \frac{1}{T}\int_0^T \Big(\alpha \mathbb{E}[P(t)]^2 + \beta \text{Var}(P(t)) \Big) dt$$

where the first term of the integrand quantifies the service capacity stability, and the second term quantifies the service capacity predictability. The coefficient $\alpha, \beta (\geq 0)$ balances the predictability and stability of $P(t)$, and the objective function reduces to the time average of $\mathbb{E}[P(t)^2]$ at $(\alpha, \beta) = (1, 1)$.

We also consider the case when the service requirements do not need to be perfectly satisfied. In such cases, we consider the optimization problem

$$\text{minimize} \quad \frac{1}{T}\int_0^T \Big(\text{Var}(P(t)) + \mathbb{E}[U(t)] + \mathbb{E}[W(t)] \Big) dt,$$

which balances service capacity variance with the penalties for not meeting the demands and/or deadlines of some jobs.
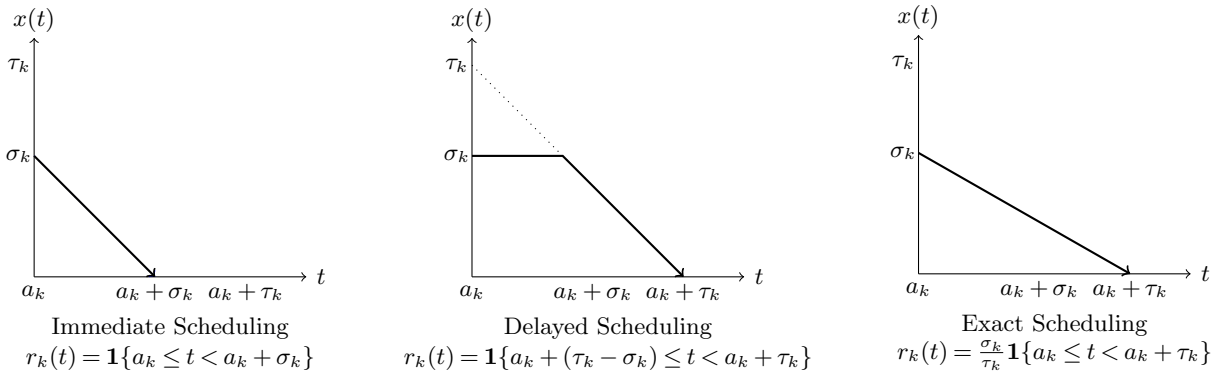
Figure 1 **Examples of distributed scheduling algorithms. The solid black lines represent the remaining demand $x(t)$ at time t. Immediate Scheduling works by serving jobs at full rate upon arrival. Delayed Scheduling works by serving at full rate with a delay that is equal to its laxity $a + \tau - \sigma$. Exact Scheduling works by throttling service to a constant rate σ/τ so that all jobs are completed exactly at its deadline.**

Motivating examples. The general model we have defined is meant to give insight into the design trade-offs that happen in applications with dynamic capacity, *e.g.* electric vehicle charging, cloud content providers. Importantly, in this paper we are not trying to model a specific application, rather we are exploring design trade-offs using a simple, general model. However, to highlight the connection to our motivating examples, consider first the case of electric vehicle charging. In this case, each job $k \in \mathcal{V}$ corresponds to an electric vehicle with an arrival time a_k, an energy demand σ_k, and a sojourn time τ_k. At each time t, the charging station draws $P(t) = \sum_{k \in \mathcal{V}} r_k(t)$ amount of power from the grid to provide each vehicle k with a charging rate of $r_k(t)$. When doing so, stable resource usage is highly desirable because fluctuations and large peaks in $P(t)$ can strain the grid and result in a high peak charge for station operators. Moreover, predictable resource usage is also important when purchasing energy from the day-ahead market, whose price is lower and less volatile than that of the real-time market. Note that our model assumes $P(t)$ is unbounded and, thus corresponds to a setting where there are more charging stations than arriving cars.

In the case of cloud content providers, each job $k \in \mathcal{V}$ corresponds to a task (requested to the cloud or data centers) with an arrival time a_k, a work requirement σ_k, and an allowable waiting
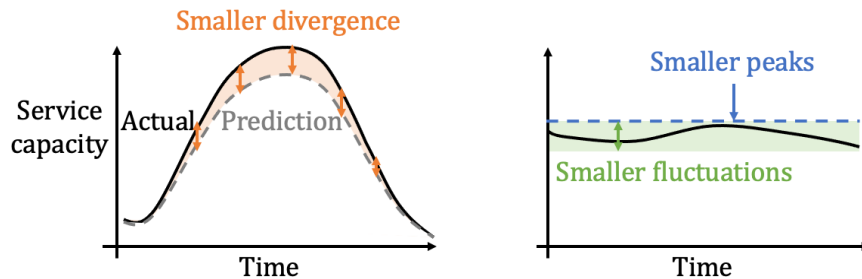
Figure 2 **Illustration of the design objectives. The left plot illustrates the predictability of the service capacity, and the right plot illustrates the stability.**

time τ_k. The service system works on job k with speed $r_k(t)$ using $P(t) = \sum_{k \in \mathcal{V}} r_k(t)$ number of computers (or amount of power). Here again, a good estimate of the future resource use enables the cloud users to reserve resources through a long-term contract, whose price is lower and less volatile than that of a short-term contract, suggesting the benefit of having predictable resource use. Note that our model considers the case where $P(t)$ is unbounded and, thus, the data center has enough capacity to avoid congestion, *i.e.* is in low utilization. Such periods are common, since data centers often operate at utilizations as low as 10% [36]. For future work, it is important to study how to manage congested periods by considering an upper bound on $P(t)$.

In this paper, we primarily focus on the cases when the arrival times, demands, and sojourn times are all available to the scheduler upon the arrival of each job. Such cases are common in many scheduling problems and modern applications operating on an increasingly smarter infrastructure [6, 14, 15, 17, 33, 37, 38]. For example, in the electric vehicle charging testbed [15], the users input the service request (σ_k, and τ_k) through a control panel upon arrival. In cloud computing, the demands can be estimated from the past, and deadlines are determined by operational/performance requirements [38]. Beyond the case of our primary focus, there are also situations when the information on demands and deadlines cannot be accessed for some or all jobs. For such cases, we discuss the algorithm to be used and its performance analysis in Section 4.1.

3. Maximizing predictability under stationary job arrivals

In this section, we characterize optimal distributed scheduling policies in a wide range of objectives when job arrivals are stationary, starting with the simplest and moving toward the most complex. To begin, we define each setting and pose the scheduler design problems as constrained functional optimizations (Section 3.1). Then, we focus on strict service requirements and show that Exact Scheduling minimizes the stationary variance of the service capacity (Section 3.2). Relaxing the demand requirements, we show that a variation of Exact Scheduling minimizes the weighted sum of the stationary variance of service capacity and the penalty for unsatisfied demand (Section 3.3). Relaxing the deadline requirements, we show that a different variation of Exact Scheduling minimizes the weighted sum of both the stationary variance of service capacity and the penalty for demand extension (Section 3.4). Finally, we consider the case when both the demand and deadline requirements are relaxed (Section 3.5) and show that the optimal policy can be constructed from an integration of the above optimal policies. It is interesting that all optimal algorithms admit closed-form expressions, which provide clear interpretations and insights regarding the optimal trade-offs between reducing service capacity variability, satisfying the demands, and meeting deadlines. Moreover, they are also highly scalable and easy to implement.

3.1. Problem formulation

We study a setting where the time horizon is infinite, *i.e.* $T = \infty$, and the arrival process is an independently marked *stationary* Poisson point process. The intensity function of the process is

$$\Lambda(a, \sigma, \tau) = \Lambda f(\sigma, \tau), \qquad\qquad a \in \mathcal{T}, (\sigma, \tau) \in S,$$

where $\Lambda(a, \sigma, \tau)$ takes the same value for different values of a given fixed σ, τ. We first consider the case when the unit cost for unsatisfied demands δ_k and that for deadline extensions ϵ_k are deterministic and homogeneous among different jobs, *i.e.* $\delta_k = \delta$, $\epsilon_k = \epsilon$ for any $k \in \mathcal{V}$.[3] We consider distributed scheduling policies of the form

$$r_k(t) = u(x_k(t), y_k(t)) \geq 0 \tag{4}$$

where $u : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}_+$ is a non-negative integrable function of the remaining demand $x_k(t)$ and the remaining time $y_k(t)$ of job k.

Scheduling policies of the form (4) can change the service rate based on the remaining time and remaining demand, pause service in the middle (we allow discontinuity in u), and stop before the full completion of jobs in the cases of soft demands. This policy assumes that the system can access the information about the service demands and deadlines. This assumption holds, for example, in electric vehicle charging systems [15]. The policy is also distributed in the sense that service rate of a job is determined using only its own information. We study policies of the form (4) assuming a situation where there is enough capacity available to satisfy the demand, and so the focus is on determining the optimal service rate for the jobs in a distributed manner.

In the special case when Immediate Scheduling policy is used, the system becomes $M/G/\infty$ queue.[4] More generally, under any policy of the form (4), the remaining job process $\{(x_k(t), y_k(t)) : k \in \mathcal{V}, a_k \leq t\}$ can be represented as a point process in a two-dimensional space of remaining times and remaining demands [35]. As $t \to \infty$, the process $\{(x_k(t), y_k(t)) : k \in \mathcal{V}, a_k \leq t\}$ converges to a stationary point process whose distribution is determined by the job profiles and scheduling policy. Moreover, it is a Poisson process in the space with mean measure $\lambda(x, y)$ satisfying:[5]

$$0 = \frac{\partial}{\partial x}(\lambda(x, y) u(x, y)) + \frac{\partial}{\partial y} \lambda(x, y) + \Lambda f(x, y). \tag{5}$$

The above equation is also known as the continuity equation and can be derived from the movement and conservation of density of the Point Process [20, 39]. The movement of each point (x, y) has velocity $-u(x, y)$ in the x-dimension and velocity -1 in the y-dimension because its remaining demand is reduced by $u(x, y)$ per unit time, and its remaining time is reduced by 1 per unit time. The conservation of density states that the flow inward (of existing jobs) and new arrivals minus flow outward through the surface of a region sum up to be zero.

Because the remaining job process becomes stationary as $t \to \infty$, the distribution of $P(t)$ also becomes stationary, *i.e.* $P(t) \to P$ in distribution. Moreover, its stationary mean $\mathbb{E}[P]$ is determined only by the total service provided. For example, in the special case when the demand constraints are to be strictly satisfied, we have $\mathbb{E}[P] = \Lambda \mathbb{E}[\sigma]$. In a more general setting, the stationary mean is given in the following proposition.

Proposition 1. *Consider a service system with stationary Poisson arrivals with intensity measure $\Lambda f(\sigma, \tau)$ and a distributed scheduling policy of the form (4). Let us define $\hat{\sigma}(\sigma, \tau)$ to be the total service a job with demand σ and a sojourn time τ receives.[6] The stationary mean of $P(t)$ is given by*

$$\mathbb{E}[P(t)] = \Lambda \mathbb{E}[\hat{\sigma}(\sigma, \tau)].$$

---

[3] This assumption is relaxed in Corollary 1.

[4] In general, the system under policy (4) may be different from $M/G/\infty$ queue because $M/G/\infty$ requires the service rate to be constant.

[5] We use (x, y) to denote the coordinate in the two dimensional space of remaining demands and remaining times and $(x_k(t), y_k(t))$ to denote a point (job profile) in the space at time t.

[6] Here, σ, τ are random variables, and $\hat{\sigma}(\sigma, \tau)$ is the output of the function with input (σ, τ). So $\hat{\sigma}(\sigma, \tau)$ is also a random variable.

We present a proof of Proposition 1 in Appendix B. Alternatively, it can also be derived from classical queueing results such as Little's Law and the Brumelle's formula [40, Chapter 3, eq. (3.2.1)]. From Proposition 1, $\mathbb{E}[P(t)]$ only depends on Λ and $\mathbb{E}[\hat{\sigma}(\sigma, \tau)]$, but it takes the same value for an arbitrary choice of policy u in (4) with identical $\hat{\sigma}(\sigma, \tau)$. In the special case when the demand constraint (3) is strictly enforced, $\mathbb{E}[P(t)] = \Lambda\mathbb{E}[\sigma]$ equals the expected value of total service demand arriving per unit time. This property suggests that the stability of the service capacity, measured using $\lim_{T\to\infty}\int_0^T \mathbb{E}[P(t)]^2 dt$, does not depend on the choice of policy u given $\mathbb{E}[\hat{\sigma}(\sigma, \tau)]$. In other words, there is no tradeoff between maximizing stability vs. predictability.

Accordingly, we consider minimizing the stationary variance of $P(t)$ under strict service (demand and deadline) constraints, soft demand constraints, soft deadline constraints, and soft demand and deadline constraints. In the case of *strict service (demand and deadline) constraints*, we consider the following optimization problem:

$$\underset{u:(1)(2)(3)(4)(5)}{\text{minimize}} \quad \text{Var}(P), \tag{6}$$

where the optimization variable is taken over the set of distributed policies (4) subject to the service rate constraints (1), the demand constraints (2), and the deadline constraints (3). Here, $\text{Var}(P)$ is a functional of u and $\lambda(\sigma, \tau)$, where $\lambda(\sigma, \tau)$ satisfies (5).

In the case of *soft demand constraints*, we relax the demand constraints (2) into paying penalty $\delta_k = \delta$ for each unit of unsatisfied demands and consider balancing the service capacity variance and the penalties due to unsatisfied demands:

$$\underset{u:(1)(3)(4)(5)}{\text{minimize}} \quad \text{Var}(P) + \mathbb{E}[U]. \tag{7}$$

In the case of *soft deadline constraints*, we relax the deadline constraints (3) into paying penalty ϵ for each unit of deadlines extensions and consider balancing the service capacity variance and the penalties due to deadline extensions:

$$\underset{u:(1)(2)(4)(5)}{\text{minimize}} \quad \text{Var}(P) + \mathbb{E}[W]. \tag{8}$$

In the case of *soft demand and deadline constraints*, we relax both the demand and deadline requirements (2) and (3) into paying δ for each unit of unsatisfied demands and ϵ for each unit of deadline extensions. We consider balancing the service capacity variance and the penalties due to unsatisfied demands and deadline extensions:

$$\underset{u:(1)(4)(5)}{\text{minimize}} \quad \text{Var}(P) + \mathbb{E}[U] + \mathbb{E}[W]. \tag{9}$$

Finally, we consider the most general setting, when the penalties for unsatisfied demands and deadlines are heterogeneous among jobs. To account for this heterogeneity, we consider distributed scheduling policies of the form

$$r_k(t) = \bar{u}(x_k(t), y_k(t), \delta_k, \epsilon_k) \geq 0. \tag{10}$$

Under any policy of the form (10), the remaining job profiles in the system $\{(x_k(t), y_k(t), \delta_k, \epsilon_k) : k \in \mathcal{V}, a_k \leq t\}$ can be represented as a point process in the 4-dimensional space of remaining times, remaining demands, unit costs for unsatisfied demand, and unit costs for deadline extension. This point process converges to a stationary Spatial Poisson Point Process with an intensity function $\lambda(x, y, \delta, \epsilon)$ satisfying

$$0 = \frac{\partial}{\partial x}(\lambda(x, y, \delta, \epsilon)\bar{u}(x, y, \delta, \epsilon)) + \frac{\partial}{\partial y}\lambda(x, y, \delta, \epsilon) + \Lambda f(x, y)g(\delta)h(\epsilon). \tag{11}$$

This leads to the following optimization problem:

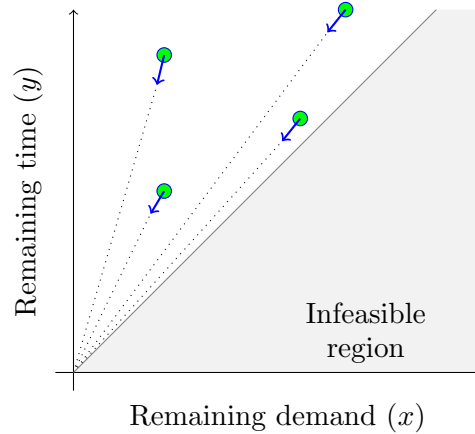$$\underset{\bar{u}:(1)(10)(11)}{\text{minimize}} \quad \text{Var}(P) + \mathbb{E}[U] + \mathbb{E}[W]. \tag{12}$$

Figure 3   **Exact scheduling depicted in the space of remaining demand x and remaining time y.**

3.2.   Strict service (demand and deadline) requirements

We first consider the case of strict service requirements and show a closed-form characterization of the optimal algorithm that minimizes the stationary variance $\mathrm{Var}(P)$. To do so, it can be shown that service rates contribute to the service capacity variance in a quadratic form (see Lemma 2, presented later). Thus, having a large value in the service rate, *i.e.* $u(x,y)$ taking large values for some (x,y), results in disproportionately more service capacity variance. This observation suggests that having a flat service rate may achieve small variance. One such policy is *Exact Scheduling*,

$$u(x,y) = \begin{cases} \dfrac{x}{y}, & \text{if } y > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

which works by finishing each job *exactly* at its deadline using a constant service rate (Figure 3). It is also highly scalable because it is distributed, and it does not require much computation, memory use, communication, or synchronization. Although existing literature has analyzed its performance in various settings [3, 12, 13, 41], no work has shown its optimality conditions in either centralized or distributed settings. In this section, we show that Exact Scheduling minimizes the stationary service capacity variance under strict demand and deadline constraints.

Theorem 1. *Exact Scheduling* (13) *is the optimal solution of* (6) *and achieves the optimal value*[7]

$$\mathrm{Var}(P) = \Lambda \mathbb{E}\left[\frac{\sigma^2}{\tau}\right].$$

Theorem 1 shows that the optimal policy for minimizing variance is to keep a constant service rate at all times. Therefore, when considering strict demands and deadlines, the optimal policy is to have a flat service rate across its sojourn time τ_k. Additionally, Theorem 1 shows the achievable performance improvement by controlling the service capacity using distributed algorithms. If no control is applied, then $r_k(t) = \mathbf{1}\{t \in [a_k, a_k + \sigma_k)\}$, and the stationary mean and variance of $P(t)$ is $\mathbb{E}(P) = \mathrm{Var}(P) = \Lambda \mathbb{E}[\sigma]$ By performing a distributed service capacity control, the stationary variance can be reduced by

$$\Lambda \mathbb{E}\left[\frac{\sigma(\tau - \sigma)}{\tau}\right] \in \left[0, \Lambda \mathbb{E}[\sigma]\right]$$

where $\tau - \sigma$ is a slack time (the amount of time left at job completion if a job is served at its maximum service rate).

---

[7] Observe that Λ is the cumulative arrival rate.

Next, we present the proof of Theorem 1. To circumvent the complex constraints of (6), we first provide a lower bound on its optimal solution by relaxing the class of control policies into

$$r_k(t) = v(\sigma_k, \tau_k, y_k(t)) \quad k \in \mathcal{V}, \tag{14}$$

and solve the optimization problem

$$\underset{v:(1)(2)(3)(14)}{\text{minimize}} \quad \text{Var}(P). \tag{15}$$

Notice that any policy that can be realized by u in (4) can also be realized by v in (14), but a policy that can be realized by v may not necessarily be realized by u. Thus, policy v is more general than u, and the constraint set of (6) is contained in the constraint set of (15). Consequently, the optimal value of (15) lower-bounds that of (6). Therefore, given the optimal solution of (15), if the solution of (15) (given in the next lemma) is also achievable by a control policy u of the form (4), it must be the optimal solution of (6) as well.

Lemma 1. *The optimal solution of* (15) *is*

$$v(\sigma, \tau, y) = \frac{\sigma}{\tau}\mathbf{1}\{y > 0\}, \tag{16}$$

and it yields the optimal value

$$\text{Var}(P(t)) = \Lambda\mathbb{E}\left[\frac{\sigma^2}{\tau}\right].$$

To show Lemma 1, we use the following property of the system: since the service rate of a job only depends on the property of that job, its impact on $\text{Var}(P)$ can be computed by integrating along the trajectory of a job over its distribution $\Lambda f(\sigma, \tau)$ [35].[8] In particular, the following relation holds.

Lemma 2. *The mean and variance of $P(t)$ under the policy* (14) *are given by*

$$\mathbb{E}[P] = \int_{(\sigma,\tau)\in S} \int_0^\tau v(\sigma, \tau, y)\Lambda f(\sigma, \tau)dyd\sigma d\tau$$

$$\text{Var}(P) = \int_{(\sigma,\tau)\in S} \int_0^\tau v(\sigma, \tau, y)^2\Lambda f(\sigma, \tau)dyd\sigma d\tau.$$

Lemma 2 can be obtained from Campbell's theorem (see Appendix A). Now we are ready to prove Lemma 1.

Proof of Lemma 1. The demand constraints (2) and the deadline constraints (3) leads to

$$\int_0^\tau v(\sigma, \tau, y)dy = \sigma, \quad (\sigma, \tau) \in S. \tag{17}$$

The objective function (15) satisfies

$$\text{Var}(P) = \int_{(\sigma,\tau)\in S} \int_0^\tau v(\sigma, \tau, y)^2\Lambda f(\sigma, \tau)dyd\sigma d\tau \tag{18}$$

$$= \int_{(\sigma,\tau)\in S} \left\{\int_0^\tau v(\sigma, \tau, y)^2 dy\right\}\Lambda f(\sigma, \tau)d\sigma d\tau \tag{19}$$

$$\geq \int_{(\sigma,\tau)\in S} \left\{\frac{\sigma^2}{\tau}\right\}\Lambda f(\sigma, \tau)d\sigma d\tau. \tag{20}$$

---

[8] This is a restatement of Brumelle's formula [42] from queueing theory for systems with infinitely many servers with time-varying rates.

Here, equality (18) is due to Lemma 2. Inequality (20) is due to (17) and the Holder's inequality, *i.e.* for any fixed (σ, τ),

$$\left(\int_0^\tau v(\sigma, \tau, y)^2 dy\right)^{1/2} \left(\int_0^\tau 1 dy\right)^{1/2} \geq \int_0^\tau v(\sigma, \tau, y) dy = \sigma,$$

where $v(\sigma, \tau, y) \geq 0$. Alternatively, it can be verified that (20) can be attained with equality when v equals (16). Therefore, (16) is the optimal solution of (15).   Q.E.D.

Lemma 1 considers the optimal scheduler among policies of the form $v(\sigma, \tau, y)$, which takes a more general form than $u(x, y)$ with the same objective function. Interestingly, the optimal scheduler in Lemma 1 does not use the additional freedom given by $v(\sigma, \tau, y)$ and can be represented by the form $u(x, y)$. This indicates that accounting for job arrival times by considering a more complex form of scheduler $v(\sigma, \tau, y)$ does not increase the system performance. Now, we can prove Theorem 1 using Lemma 1.

Proof of Theorem 1.   Recall that the optimal solution of (15) is the policy (16). Under the policy (16), the ratio between its remaining demand $x(t)$ and remaining time $y(t)$ are constant for any $t \in [a, a + \tau]$. Therefore, (16) can be realized using policies of the form (4). Because the optimal value of (15) is a lower bound on that of (6), the optimal solution of (15)—Exact Scheduling—is also the optimal solution of (6).   Q.E.D.

In fact, the same property also holds for the optimization problems (7), (8), and (9). This property allows us to derive their closed-form solutions.

3.3.   Soft demand requirements

The previous section shows the optimal algorithm under strict service constraints. In this section, we relax the assumption of strict service constraints and characterize the optimal algorithm under soft demand constraints. Specifically, we consider the setting of (7), where the system does not need to satisfy all demands but needs to pay a penalty δ for each unit of unsatisfied demands. The resulting optimal algorithm is a variation of Exact Scheduling with an additional rate upper-bound:

$$u(x, y) = \begin{cases} \dfrac{x}{y}, & \text{if } \dfrac{x}{y} \leq \dfrac{\delta}{2} \text{ and } y > 0, \\ \dfrac{\delta}{2}, & \text{if } \dfrac{x}{y} > \dfrac{\delta}{2} \text{ and } y > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

This policy essentially imposes a threshold (an upper bound) of $\delta/2$ on the service rate: jobs whose ratio σ/τ is above threshold $\delta/2$ are served at a constant rate $\delta/2$ *until its deadline*; jobs whose ratio σ/τ is below this threshold are served according to Exact Scheduling. In other words, a job k receives its full service demand only if $\sigma_k/\tau_k \leq \delta/2$.

Theorem 2. *The policy* (21) *is the optimal solution of* (7) *and achieves the optimal value*

$$\text{Var}(P) + \mathbb{E}[U] = \mathbb{E}\left[\frac{\sigma^2}{\tau} \mathbf{1}\left\{\frac{\sigma}{\tau} \leq \frac{\delta}{2}\right\} + \delta\left(\sigma - \frac{\delta\tau}{4}\right) \mathbf{1}\left\{\frac{\sigma}{\tau} > \frac{\delta}{2}\right\}\right] \Lambda. \tag{22}$$

A proof of Theorem 2 is given in Appendix C. Theorem 2 shows the performance improvement gained by relaxing the demand requirements. Recall from Theorem 1 that the average cost per unit job arrival is $\mathbb{E}[\sigma^2/\tau]$ if all demands must be satisfied. If the system does not need to satisfy all demand requests, then the average cost for jobs satisfying $\sigma/\tau > \delta/2$ can be reduced from $\mathbb{E}[\sigma^2/\tau]$ to $\mathbb{E}[\delta(\sigma - (\delta\tau/4))]$. And the portion of such jobs is given by $\mathbb{E}[\mathbf{1}\{\sigma/\tau > \delta/2\}]$. The optimal policy (21) is also simple and easy to implement. Despite the convenience and wide use of simple thresholding policies in practice, to the best of our knowledge, its optimality results and the optimal choice of thresholding values on rate have not been proposed in the existing literature.

3.4. Soft deadline requirements

The previous section shows the optimal algorithm under soft demand requirements. In this section, we relax the deadline requirements instead and characterize the optimal distributed algorithm. Specifically, we consider the setting of (8), where the system needs to pay penalty ϵ for each unit of deadline extensions. Although scheduling problems with deadline extension (tardiness) are often NP-hard [43, 44], here we present the optimal distributed algorithm in closed-form for infinite time horizon. This is done by first relaxing the original optimization problem into the ones that can be solved analytically and then showing the feasibility and optimality of the obtained solution in the original problem. The resulting optimal algorithm is a variation of Exact Scheduling with deadline extensions:

$$u(x,y) = \begin{cases} \dfrac{x}{y} & \text{if } \dfrac{x}{y} \leq \sqrt{\epsilon} \text{ and } y > 0 \\ \sqrt{\epsilon}\,\mathbf{1}\{x > 0\} & \text{otherwise.} \end{cases} \qquad (23)$$

Similarly to (21), this policy essentially sets a threshold (an upper bound) $\sqrt{\epsilon}$ on the service rate: jobs with the ratio above threshold $\sqrt{\epsilon}$ are served according to Equal Service of rate $\sqrt{\epsilon}$ *until it finishes*, jobs with the ratio below threshold $\sqrt{\epsilon}$ are served according to Exact Scheduling. In other words, the deadline of job k is extended only if $\sigma_k/\tau_k > \sqrt{\epsilon}$.

Theorem 3. *The policy* (23) *is the optimal solution of* (8) *and achieves the optimal value*

$$\mathrm{Var}(P) + \mathbb{E}[W] = \mathbb{E}\left[\frac{\sigma^2}{\tau}\mathbf{1}\left\{\frac{\sigma}{\tau} \leq \sqrt{\epsilon}\right\} + \left(2\sqrt{\epsilon}\sigma - \epsilon\tau\right)\mathbf{1}\left\{\frac{\sigma}{\tau} > \sqrt{\epsilon}\right\}\right]\Lambda. \qquad (24)$$

Theorem 3 shows the performance improvement by relaxing the deadline requirements. Theorem 1 states that, if all deadlines must be satisfied, then the average cost per unit job arrival is $\mathbb{E}\left[\sigma^2/\tau\right]$. By allowing deadline extensions, the average cost of jobs satisfying $\sigma/\tau > \sqrt{\epsilon}$ can be reduced from $\mathbb{E}\left[\sigma^2/\tau\right]$ to $\mathbb{E}\left[(2\sqrt{\epsilon}\sigma - \epsilon\tau)\right]$, and the portion of such jobs is given by $\mathbb{E}\left[\mathbf{1}\{\sigma/\tau > \sqrt{\epsilon}\}\right]$. Moreover, service capacity variance and penalties for deadline extension are optimally balanced when jobs whose deadline extension penalties are smaller than σ/τ, i.e. $\sigma/\tau > \sqrt{\epsilon}$, are served with deadline extension.

3.5. Soft demand and deadline requirements

The previous sections show the optimal algorithms under soft demand requirements and soft deadline requirements. In this section, we relax both demand and deadline requirements and characterize the optimal distributed algorithm. Specifically, we consider the setting of (9) where the system needs to pay penalty δ for each unit of unsatisfied demands and penalty ϵ for each unit of deadline extensions. This setting recovers all previous settings as special cases.[9]

Recall from previous sections that, under soft demand requirements, the optimal policy uses a constant service rate and rejects partial demand requests only if $\sigma/\tau > \delta/2$. Meanwhile, under soft deadline requirements, the optimal policy uses a constant service rate and extends the deadline only if $\sigma/\tau > \sqrt{\epsilon}$. These two special cases motivate us to combine the policies (13), (21), and (23) as follows:

$$u(x,y) = \begin{cases} \dfrac{x}{y} & \text{if } y > 0 \text{ and } \dfrac{x}{y} \leq \min\left\{\dfrac{\delta}{2}, \sqrt{\epsilon}\right\} \\ \dfrac{\delta}{2} & \text{if } y > 0 \text{ and } \dfrac{x}{y} > \dfrac{\delta}{2} \text{ and } \dfrac{\delta}{2} \leq \sqrt{\epsilon}, \\ \sqrt{\epsilon}\,\mathbf{1}\{x > 0\} & \text{otherwise} \end{cases} \qquad (25)$$

[9] For sufficiently large δ, this setting recovers the case of strict demand requirements. For sufficiently large ϵ, this setting recovers the case of strict deadline requirements. For sufficiently large δ and ϵ, this setting recovers the case of strict demand and deadline requirements.
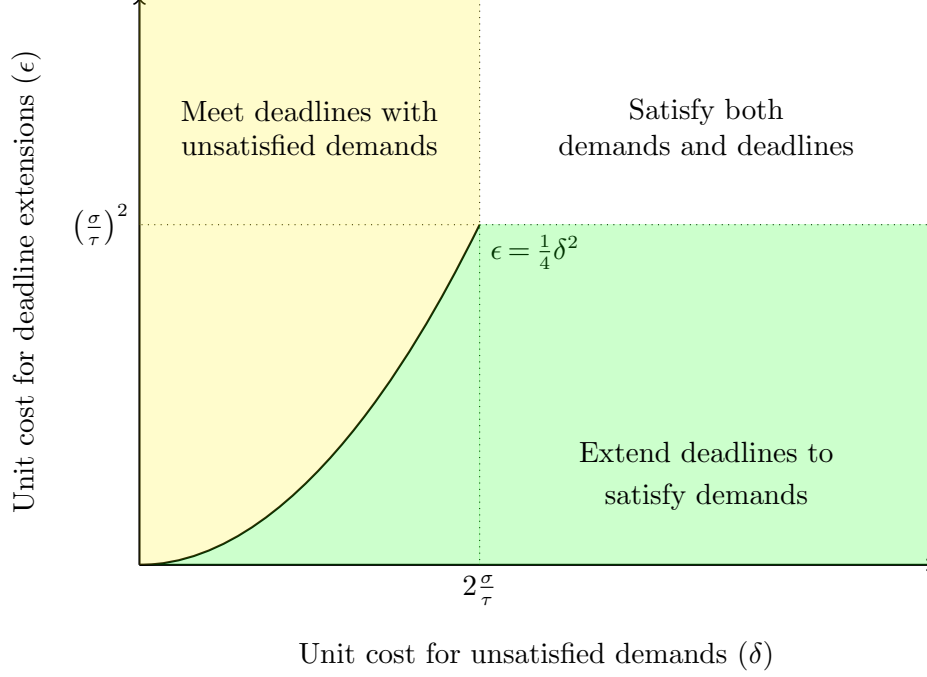
Figure 4   The decision space of the optimal policy for (9). **For job profiles with a service demand σ, a sojourn time τ, and costs (δ, ϵ), the optimal policy performs either one of the following using constant service rates: satisfy both demands and deadlines (white region), meet deadlines with unsatisfied demand (yellow region), or satisfy the demand by extending the deadline (green region).**

The policy uses three strategies depending on different regimes of job states and penalties: high penalties regime, low demand penalty regime, and low deadline penalty regime. These regimes are illustrated in Figure 4 as the white, green, and yellow regions, respectively.

• *High penalties regime.* When $\min(\delta/2, \sqrt{\epsilon}) > \sigma/\tau$, it is less costly to satisfy the service requirements than paying penalties for unsatisfied demands or deadlines. So, the best strategy is to satisfy both demands and deadlines optimally using Exact Scheduling (13).

• *Low demand penalty regime.* When $\delta/2 \leq \sqrt{\epsilon}$, the penalties for unsatisfied demands is smaller than that of deadline extensions, so the best strategy is to meet all deadlines optimally with potentially unsatisfied demands using the policy (21).

• *Low deadline penalty regime.* When $\delta/2 > \sqrt{\epsilon}$, the penalties for deadline extension is smaller than that of unsatisfied demands, so the best strategy is to satisfy demands optimally with potential deadline extensions using the policy (23).

From above, the policy (25) generalizes the optimal algorithms in Sections 3.2-3.4, and we term it *Generalized Exact Scheduling.* The following theorem states its optimality condition.

Theorem 4. *The policy (25) is the optimal solution of (9) and achieves the optimal value*

$$\mathrm{Var}(P) + \mathbb{E}[U] + \mathbb{E}[W] = \tag{26}$$
$$\mathbb{E}\left[\frac{\sigma^2}{\tau}\mathbf{1}\left\{\frac{\sigma}{\tau} \leq \min\left\{\frac{\delta}{2}, \sqrt{\epsilon}\right\}\right\} + \delta\left(\sigma - \frac{\delta\tau}{4}\right)\mathbf{1}\left\{\frac{\sigma}{\tau} > \frac{\delta}{2}, \frac{\delta}{2} \leq \sqrt{\epsilon}\right\} + \left(2\sqrt{\epsilon}\sigma - \epsilon\tau\right)\mathbf{1}\left\{\frac{\sigma}{\tau} > \sqrt{\epsilon}, \frac{\delta}{2} > \sqrt{\epsilon}\right\}\right]\Lambda.$$

Theorem 4 shows when one should extend the deadline to satisfy the demand or let the job depart at its deadline with unsatisfied demands. Moreover, Generalized Exact Scheduling is also optimal for a more general problem (12), when the unit costs for unsatisfied demands and deadline extensions are allowed to be heterogeneous.

Corollary 1. *The optimal solution of* (12) *is*

$$\bar{u}(x,y,\delta,\epsilon) = \begin{cases} \dfrac{x}{y} & \text{if } y > 0 \text{ and } \dfrac{x}{y} \leq \min\left\{\dfrac{\delta}{2}, \sqrt{\epsilon}\right\} \\[2ex] \dfrac{\delta}{2} & \text{if } y > 0 \text{ and } \dfrac{x}{y} > \dfrac{\delta}{2} \text{ and } \dfrac{\delta}{2} \leq \sqrt{\epsilon} \\[2ex] \sqrt{\epsilon}\,\mathbf{1}\{x > 0\} & \text{otherwise} \end{cases}.$$

Corollary 1 is an immediate consequence of Theorem 4.

4. Performance degradation inherent to availability of job information

Given our focus on distributed algorithms, we now investigate how much performance degrades in comparison to centralized algorithms. This investigation also includes practically important questions such as: is there is any middle ground between centralized and distributed algorithms in which scalability and close-to-centralized performance can be achieved simultaneously? What can be done if the information on the service requirements (demands and/or deadlines) is missing? How much does the performance degrade due to such missing information? In this section, we answer these questions using both numerical experiments and theory.

Specifically, we compare the performance of optimal offline algorithms, centralized online optimization, particularly-centralized algorithms, online distributed algorithms using actual electric vehicle charging data from the Caltech Testbed [15] and synthetic data drawn from varying arrival distribution (Section 4.1). The results show that Generalized Exact Scheduling has superior performance when compared to other distributed algorithms. This performance is achieved by systematically optimizing service capacity to find the right balance between service capacity variance and the penalties for unsatisfied demands or deadlines. Moreover, the results in the testbed also suggest that Generalized Exact Scheduling performs well beyond the case of Poisson arrivals, under which Exact Scheduling is optimal, such as in the charging of electric vehicles. Then, we derive bounds on the cost of the optimal centralized policy and use these bounds to characterize the performance degradation of the optimal distributed algorithm (Section 4.2). When it comes to deriving performance bounds, there is no standard technique in queueing to derive the performance limits of centralized policies in this setting. Instead, we borrow tools from optimal control and provide an upper bound on the performance. Finally, we present a proof of the upper bound, which is potentially useful for providing performance degradation bounds for policies in other settings as well (Section 4.2.1).

4.1. Empirical evaluation

To evaluate the performance of the proposed algorithms (Generalized Exact Scheduling), we compare its performance with existing scheduling algorithms using both a case study of an electric vehicle charging testbed and synthetic data of varying arrival distributions.

4.1.1. System and data We develop a trace-driven simulation using real data from an electric vehicle charging testbed [15], and use a synthetic data set randomly drawn from a set of arrival distributions with varying parameters. The real data contains the arrival profiles of 92 days in 2016. A charging instance contains service requests from each electric vehicle arriving in one day. A service request of a job is defined by its arrival time, energy demand, and sojourn time. The statistics of the service requests are summarized in Table 2. The synthetic data are generated from the following set of arrival distributions. Time is discretized into the sampled time $t_1 = 0, t_2, \cdots, t_n$. Given a vector $b(i)$ of i.i.d. Bernoulli random variables with mean $p_B (\ll 1)$, the sampled time t_i is considered to have one arrival if $b(i) = 1$; and zero arrival if $b(i) = 0$. For each arrival, its service demand σ is uniformly distributed in $[\underline{\sigma}, \bar{\sigma}]$. Its sojourn time is generated from two different cases, defined by two types of arrival distributions (I and II). In distribution I, the sojourn time is given

(a) Job profiles

	Demand σ_k (kW × minutes)	Sojourn time τ_k (minutes)
Mean	$5.1 \cdot 10^2$	$3.2 \cdot 10^2$
Variance	$3.1 \cdot 10^5$	$3.9 \cdot 10^4$

(b) Instance profiles

	Total demand $\sum_{k \in \mathcal{V}} \sigma_k$ (kW × minutes)	Time horizon T (minutes)	Number of jobs $|\mathcal{V}|$
Mean	$8.4 \cdot 10^3$	$6.5 \cdot 10^2$	14.2
Variance	$2.0 \cdot 10^7$	$1.4 \cdot 10^4$	48.2

Table 2   Statistics of the electric vehicle charging instances at the testbed [15].

by $\sigma + \ell$ (additive), where ℓ is i.i.d. exponentially distributed with mean $\bar{\ell}$. In distribution II, the sojourn time is given by $\gamma\sigma$ (multiplicative), where γ is uniformly chosen from the interval $[1, \bar{\gamma}]$. In both distributions I and II, all jobs are feasible, *i.e.* $\sigma_k \leq \tau_k$ for all $k \in \mathcal{V}$. The parameters for the arrival distributions are chosen to be $p_B \in [0.1, 0.3], \underline{\sigma} = 10, \bar{\sigma} = 20, \bar{\ell} \in [10, 50], \bar{\gamma} = 3$.

4.1.2. Algorithms We compare a few standard schedulers with Generalized Exact Scheduling. The schedulers include an optimal offline policy, online fully-centralized policies, online partially-centralized policies, and online distributed policies. The details of these schedulers are defined below.

Offline optimal algorithms (centralized). To understand the best possible performance, we compare with the optimal offline algorithms. Let $A_t = \{(a_k, \sigma_k, \tau_k, x_k(t), y_k(t)) : a_k \leq t\}$ be the set that contains the information of jobs arriving no later than t. The optimal offline algorithm tells the best performance achievable given the centralized information of *all* jobs arriving in the future. The offline policy takes the form

$$r_k(t) = o(k, t, \{A_t\}_{t \in [0,T]}), \quad \forall k \in \mathcal{V}, \tag{27}$$

where $o(k, t, \cdot)$ is a deterministic mapping from the complete arrival information $\{A_t\}_{t \in [0,T]}$ to a service rate $r_k(t)$ and is allowed to vary by each job k and each time t. The offline policy (27) can use the information of all future arrivals to generate the service rate at time t and can be computed from the following optimization problem:

$$\underset{(1)(2)(3)}{\text{minimize}} \quad \frac{1}{T} \int_0^T (P(t) - \bar{P})^2 dt, \tag{28}$$

where the optimization variable is o in (27), and $\bar{P} = (1/T) \int_0^T P(t) dt$ be the time average of $P(t)$. We denote the solution of problem (28) as Optimal Offline.

This assumption on Optimal Offline is often too strong in practice: offline algorithms cannot be used when future information is hard to obtain. However, as any distributed or online algorithms can perform no better than the optimal offline algorithm, it is still useful to have its performance as a baseline. Specifically, we quantify the relative cost of any online algorithm and Optimal Offline using the ratio of the cost of the algorithm and that of Offline Optimal (with a slight abuse of notation, we denote this ratio as the *empirical competitive-ratio*[10]). This quantity is used in Figures 5, 6, 11 to evaluate the performance of different algorithms under varying arrival distribution.

---

[10] Competitive-ratio typically refers to the worst-case ratio among all possible instances, but here, we use empirical competitive-ratio to refer to the empirically realized ratio in one instance.

Fully-centralized online algorithms. We consider centralized (online) scheduling policies of the form

$$r_k(t) = c(k, t, A_t), \quad \forall k \in \mathcal{V}, \tag{29}$$

where $c(k, t, \cdot)$ is a deterministic mapping from past and present arrival information $A_t = \{(a_k, \sigma_k, \tau_k, x_k(t), y_k(t)) : a_k \leq t\}$ to a service rate $r_k(t)$ and is allowed to vary by each job k and each time t. The centralized online policy (29) can use the information of jobs that arrived by time t to generate the service rate at time t. We list below a few centralized online policies tested in this paper.

• Online Optimization (MPC). This policy essentially performs Model Predictive Control (MPC) on the objective function (28).[11] The service rates are recomputed at discrete points in time $t_0 (= 0), t_1, t_2, \cdots, t_N (= T)$, where $N = \lceil T/h \rceil$ is the discrete index for the time horizon, and the sampling interval h $(= t_{n+1} - t_n, n \in \{0, 1, \cdots, N\})$ is chosen to be sufficiently small. Let $\mathcal{V}_t = \{k \in \mathcal{V} : a_k \leq t\}$ be the set of jobs arriving by time t, and $\mathcal{N}_\ell = \{n \in \mathbb{Z}_+ : n \geq \max(0, \ell), n \leq N\}$ be the discrete time index from ℓ (≥ 0) to N. At each discrete point in time t_i, this policy finds the service rate $r_k(\tau), \tau \geq t_i$ defined from the optimization variable $\hat{r}_k(i)$,

$$r_k(\tau) = \hat{r}_k(i) \qquad\qquad \tau \in [t_i, t_{i+1}), k \in \mathcal{V}_{t_i} \tag{30}$$

and

$$
\begin{aligned}
r_k(\tau) &= \hat{r}_k(i+1) & \tau \in [t_{i+1}, t_{i+2}), k \in \mathcal{V}_{t_i} \\
r_k(\tau) &= \hat{r}_k(i+2) & \tau \in [t_{i+2}, t_{i+3}), k \in \mathcal{V}_{t_i} \\
&\vdots \\
r_k(\tau) &= \hat{r}_k(N-1) & \tau \in [t_{N-1}, t_N], k \in \mathcal{V}_{t_i}
\end{aligned}
\tag{31}
$$

which minimizes the objective function (28) subject to the rate constraint (1), demand constraint (2), and deadline constraint (3), *i.e.*

$$\underset{\hat{r}_k(i), k \in \mathcal{V}_{t_i}, i \in \mathcal{N}_i : (1)(2)(3)(30)(31)}{\text{minimize}} \quad \frac{1}{T} \int_0^T (P(t_k) - \bar{P})^2 dt, \tag{32}$$

subject to the rate constraint (1), demand constraint (2), deadline constraint (3). The optimization problem (32) computes the service rates in discrete time, which approximate the continuous-time optimal service rate for (28) for time horizon T. Since an online algorithm cannot access future job arrivals in decision making, this problem does not use the future job arrival information, *i.e.* $\mathcal{V}_{t_n}$ in (32) only contains jobs arriving by time t_n. At each time t_n, it recomputes problem (32) using the available job information in $\mathcal{V}_{t_n}$ and serves according to (30) until the next sample time, while (31) is accounted for in the optimization problem (32), $r_k(\tau), \tau \geq t_{i+1}$ is not executed. The computed service rates are optimal if no jobs arrive in the future but are only greedily optimal otherwise.

• Fair Sharing (FS). This policy equally distributes a fixed capacity among jobs. In the case of soft demand constraints, each job k is served until their deadlines according to $r_k(t) = \min\{p_{\text{FS}}/n(t), 1\}\mathbf{1}\{y_k(t) > 0\}$, where $n(t)$ is the number of unfinished jobs at time t. Under soft deadline constraints, jobs are served to their completion according to $r_k(t) = \min\{p'_{\text{FS}}/n(t), 1\}\mathbf{1}\{x_k(t) > 0\}$. In the experiment, the parameters p_{FS} and p'_{FS} are chosen to be the optimal offline values.[12]

---

[11] See [45, 46] and references therein for the use of MPC in similar scheduling problems.

[12] We choose fixed these parameters to be the same for all instances but choose these parameters to be the ones that minimize the average cost of all instances. The optimal offline values for the parameters are numerically computed from grid search. Since the optimal offline parameters are unknown in practice, the test results obtained here are optimistic.

- Earliest Deadline First (EDF). This policy allocates a fixed capacity to jobs in ascending order of their deadlines. In the case of soft demand constraints, at time t, each job k with positive remaining time $y_k(t) > 0$ receives the smaller of the remaining demand or maximum service rate, $\min\{x_k(t), 1\}$, in the order of jobs with closest deadlines, until either the fixed capacity p_{EDF} is exhausted or all jobs receive $\min\{x_k(t), 1\}$. In the case of soft deadline constraints, at time t, each unfinished job receives the smaller of the remaining demand or maximum service rate, $\min\{x_k(t), 1\}$, in the order of jobs with closest deadlines until either the fixed capacity p_{EDF} is exhausted or all jobs receive $\min\{x_k(t), 1\}$. In the experiment, p_{EDF} and p'_{EDF} are chosen to be the optimal offline values.[12]

- Least Laxity First (LLF). This policy allocates a fixed capacity to jobs in ascending order of their laxity, where laxity of job k at time t is defined as

$$\ell_k(t) := y_k(t) - x_k(t).$$

In the case of soft demand constraints, at time t, each job k with positive remaining time $x_k(t) > 0$ receives the smaller of the remaining demand or maximum service rate, $\min\{x_k(t), 1\}$, in the order of jobs with smallest laxities, until either the fixed capacity p_{LLF} is exhausted or all jobs receives $\min\{x_k(t), 1\}$. In the case of soft deadline constraints, at time t, each unfinished job receives the smaller of the remaining demand or maximum service rate, $\min\{x_k(t), 1\}$, in the order of jobs with smallest laxities until either the fixed capacity p'_{LLF} is exhausted or all jobs receives $\min\{x_k(t), 1\}$. In the experiment, p_{LLF} and p'_{LLF} are chosen to be the optimal offline values.[12]

The parameters $p_{FS}, p'_{FS}, p_{EDF}, p'_{EDF}, p_{LLF}, p'_{LLF}$ for the above algorithms are the service capacities that the system is willing to provide in their respective settings, and this capacity is shared among all unfinished jobs.

Distributed online algorithms. Recall from Section 3.1 that a distributed online policy takes the form (4). Below lists the distributed online policies tested in our experiments.

- Generalized Exact Scheduling (25). This policy recovers Exact Scheduling (13) under strict service requirements, the policy (21) under soft demands, and the policy (23) under soft deadlines.

- Immediate Scheduling. This policy schedules jobs at its maximum rate upon arrival, *i.e.* for any job $k \in \mathcal{V}$, $r_k(t) = \mathbf{1}\{x_k(t) > 0\}$.

Partially-centralized algorithms. The fully-centralized online algorithms above require the scheduler to access all service requirement information (demands and deadlines) for all jobs present in the system, whereas the distributed online algorithms use no centralized information (*i.e.* the service rate of each job is only determined using its own information). Beyond the two extreme cases of totally centralized versus totally distributed, there is the middle ground of *partially centralized* algorithms where service rates are determined mostly using the local information of each job but are allowed to access some global state variables. The design choices for partially centralized algorithms are vast, yet their potential is under-explored in the existing literature. Although a comprehensive study of such design space is beyond the scope of this paper, we have explored a few such options empirically to facilitate future discussion. As the major goal of our paper is to design scalable algorithms, we focus on near-distributed policies that only use a limited number of global variables. Such policies take the form

$$r_k(t) = pc(x_k(t), y_k(t), z(t)) \ge 0, \qquad\qquad k \in \mathcal{V} \qquad\qquad (33)$$

where $z(t)$ is a low-dimensional vector that is shared among the local schedulers for each job. For example, $z(t)$ can contain the service capacity, total remaining demands, total remaining time, number of jobs, and combinations of these quantities. The policy (33) requires much less resources in computation and communication compared to the centralized algorithms listed above. For example, Online Optimization (MPC) solves at each time step a quadratic program; EDF and LLF require jobs to be sorted. In contrast, (33) only requires the evaluation of a closed-form function and minimum communication (sending $z(t)$ to local schedulers).

We tested the following two algorithms of the form (33).

• Exact Scheduling PC. This policy performs Exact Scheduling plus minor adjustment using partially centralized (PC) variable $P(t)$. It operates as Exact Scheduling when the service capacity is higher than average but adds additional boosts in service rate otherwise, *i.e.*

$$pc(x_k(t), y_k(t), z(t)) = \begin{cases} \mu \dfrac{x_k(t)}{y_k(t)} & P(t - dt) < \bar{P} \\ \dfrac{x_k(t)}{y_k(t)} & \text{otherwise} \end{cases}$$

where dt captures the latency in communicating $P(t)$, and $\mu \geq 1$ is the factor that boosts service rate at the low service capacity regime. Empirically, the values for μ that work well for the scheduling instances that we tested range from 1.2 to 1.6.

• Equal Service. This policy offers a homogeneous service rate to all unfinished jobs. In the case of strict service requirements, it serves jobs with positive laxity at a homogeneous service rate c_{ES} and jobs with zero laxity at its maximum rate 1. The service rate under this policy is given by $r_k(t) = c_{\text{ES}}\mathbf{1}\{\ell_k(t) > 0 \text{ and } x_k(t) > 0\} + \mathbf{1}\{\ell_k(t) \leq 0 \text{ and } x_k(t) > 0\}$. In the case of soft demand constraints, it serves jobs at a homogeneous service rate c'_{ES} before its deadline, and the service rate is given by $r_k(t) = c'_{\text{ES}}\mathbf{1}\{x_k(t) > 0 \text{ and } y_k(t) \geq 0\}$. Note that this policy may not fulfill the demand of all jobs but does satisfy all deadlines. In the case of soft deadline constraints, it serves jobs at a homogeneous service rate until its completion, and the service rate is given by $r_k(t) = c''_{\text{ES}}\mathbf{1}\{x_k(t) > 0\}$. Note that this policy satisfies all demands but may extend the deadlines of some jobs. Insights from Section 3 suggest that Equal Service may perform well when the values of $c_{\text{ES}}, c'_{\text{ES}}, c''_{\text{ES}}$ are close enough to $\mathbb{E}[v^*(\sigma_k, \tau_k, y_k(t))]$, where v^* are the optimal solutions for problems (15), (112), and (123).[13] Indeed, we observed this behavior empirically. Moreover, we noticed that Equal Service is robust to small perturbations in $c_{\text{ES}}, c'_{\text{ES}}, c''_{\text{ES}}$ near their desired value $\mathbb{E}[v^*(\sigma_k, \tau_k, y_k(t))]$.
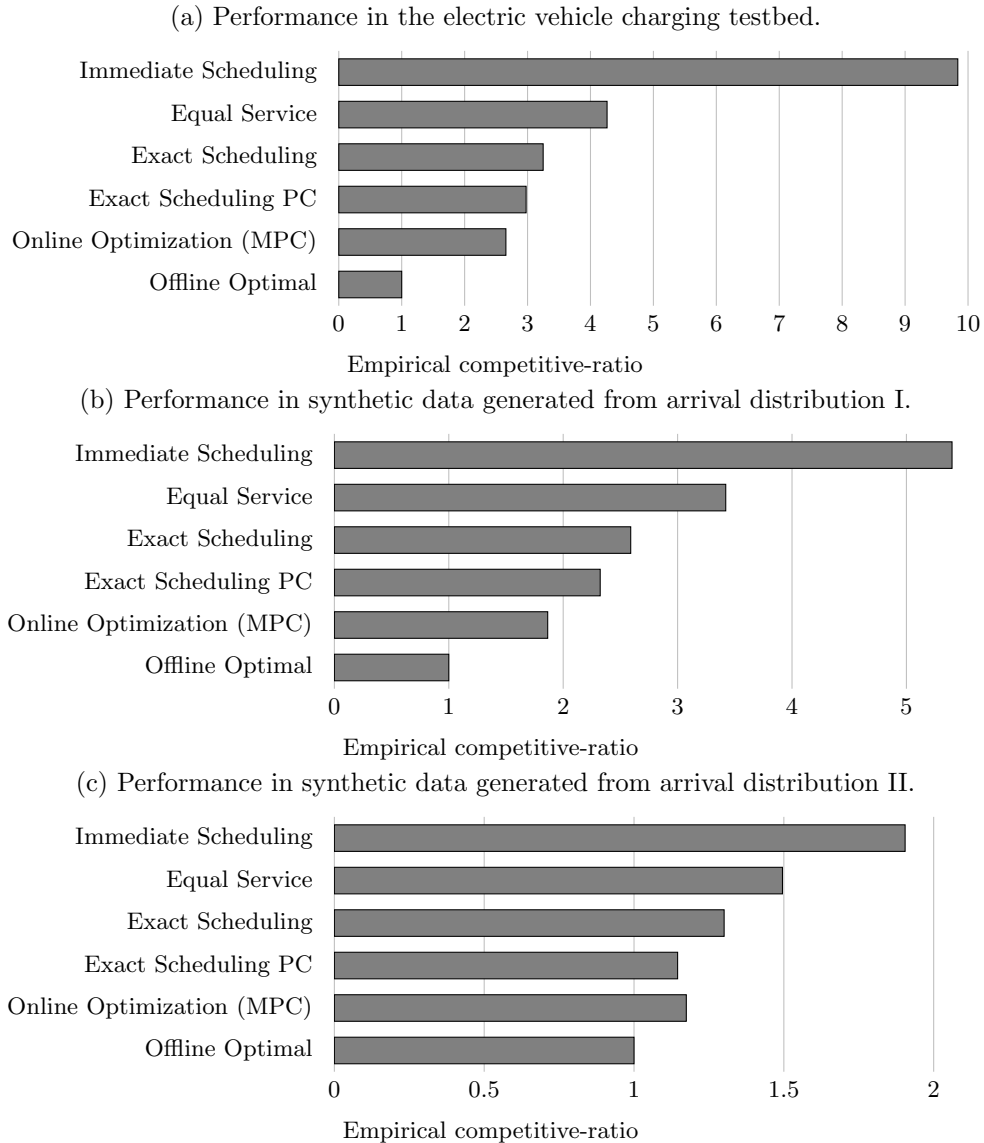
4.1.3. Fully-centralized vs partially-centralized vs distributed algorithms We tested the above algorithms in the case of strict service constraints and studied how the characteristics of the scheduling instance influence the relative performance. We also studied how different algorithms behave in the cases of soft demand constraints and deadline constraints for varying penalties of unsatisfied demands and deadlines. The results are summarized below.

Performance in strict service constraints. In the setting of strict service constraints, we can only use Exact Scheduling, Immediate Scheduling, and Equal Service, Online Optimization (MPC), and Offline Optimal because other algorithms cannot guarantee to satisfy the service requirements strictly.

For the instances in the testbed (Figure 5a), we observed a significant performance degradation from Offline Optimal to Online algorithms: online algorithms experience 50% times more cost due to the lack of future information. However, among online algorithms, there is less than 10% of cost increment from fully-centralized Online Optimization (MPC) to Exact Scheduling PC, which requires much less computational and communication resources, and another 10% cost increment from Exact Scheduling PC to fully-distributed Exact Scheduling.

For synthetic instances (Figures 5b, 5c), the performance degradation from Offline Optimal to online algorithms is much less than in the testbed, which may be attributable to the fact that the synthetic data's arrival distribution is closer to our assumptions in the arrival distribution of the optimization problems. On the other hand, the performance degradation from fully-centralized to partially-centralized to fully-distributed remains similar. Exact Scheduling PC (partially centralized) degrades from fully-centralized Online Optimization (MPC) by about 10% on average, and Exact Scheduling (fully-distributed) degrades from Exact Scheduling PC by another 10% on average. This relation holds beyond the specific arrival distribution considered in Figure 5 (see Figure 11 in the Appendix for the performance comparison in varying arrival distributions).

---

[13] Problem (112) and (123) are defined in the Appendix.

(a) Performance in the electric vehicle charging testbed.

(b) Performance in synthetic data generated from arrival distribution I.

(c) Performance in synthetic data generated from arrival distribution II.


Figure 5 **Performance comparison of algorithms under strict service (demand and deadline) constraints in the testbed [15]. The ratio of each algorithm's empirical variance to the Offline Optimal is averaged over all scheduling instances. The number of instances averaged are** 92 **in plot (a) and** 500 **in plot (b) and plot (c). The instances used here are described in Section 4.1.1. In plot (b), the arrival distribution I is set to have parameter** $\bar{\ell} = 15$**. In plot (c), the arrival distribution II is set to have parameter** $\bar{\gamma} = 2$**. For arrival distribution with different parameters from (b) and (c), the performance is shown in Figure 11 in the Appendix I.**

Performance by instance characteristics. The relative performance/cost of Exact Scheduling and others depends on the characteristics of the scheduling instance. To further investigate this dependency, we grouped instances according to its empirical competitive-ratio of Exact Scheduling and computed the average arrival rate and the demand to sojourn time ratio for each group. In general, empirical competitive-ratio improves as the number of arrivals decrease (Figure 6a) and also as average demand to sojourn time ratio increases in size (Figure 6b). These points can also be seen in Figure 10, which compares the service rates for the instance in which the Exact Scheduling performed equally well with Offline Optimal (Figure 10a) and those for the instance in which Exact Scheduling performed much worse (Figure 10b). Intuitively, sparser arrivals would require

(a) Average arrival rate vs empirical competitive-ratio

(b) Demand to sojourn time ratio vs empirical competitive-ratio

(c) Number of instances in each class, which are grouped by the range of empirical competitive-ratio
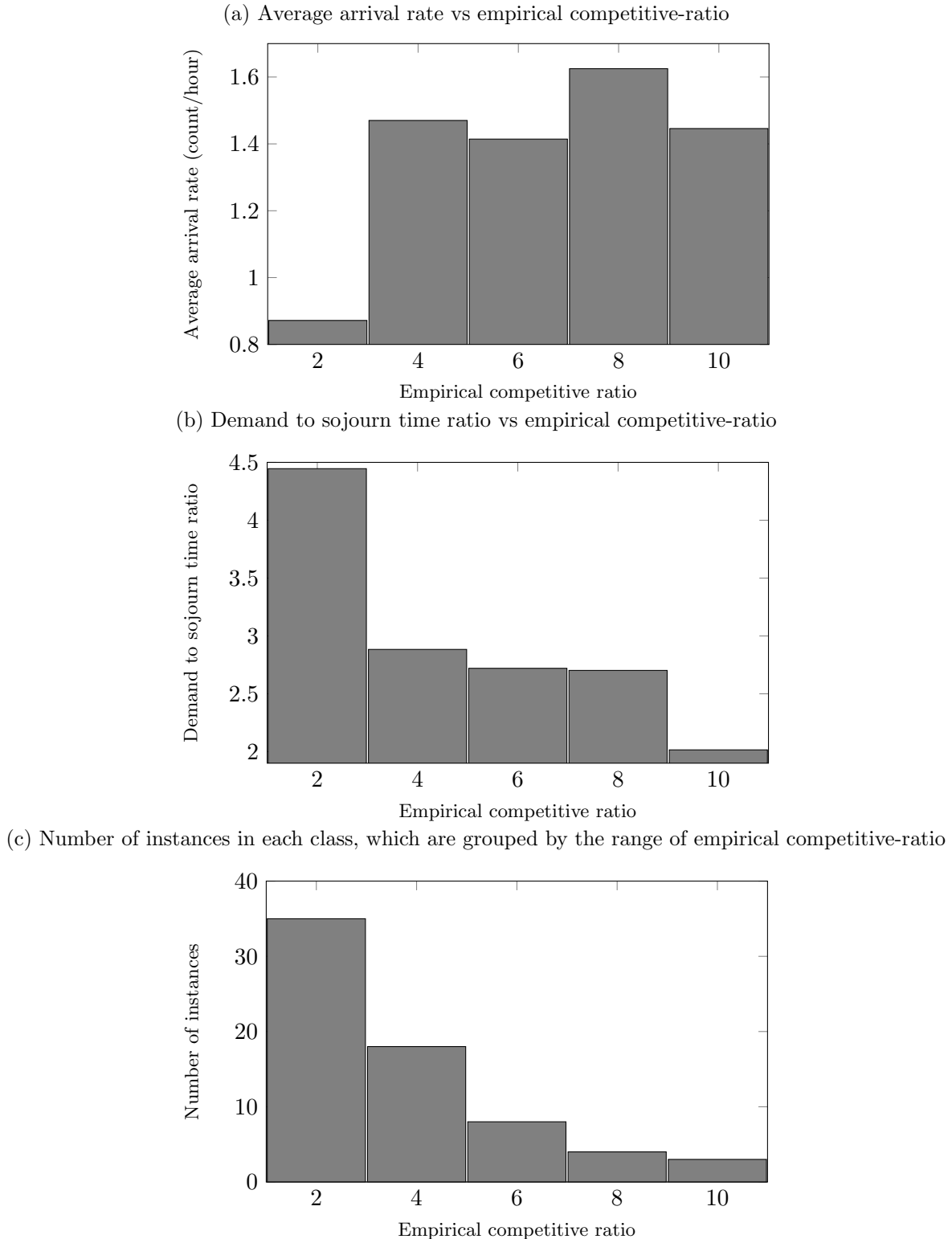
Figure 6 **Instance characteristics that allow Exact Scheduling to have comparable performance with the Offline Optimal. For each instance, the ratio between the cost of Exact Scheduling and that of Offline Optimal (denoted as the empirical competitive-ratio with a slight abuse of notation[10]) is computed. Based on this ratio, instances are grouped into 5 classes, each containing instances for which the empirical competitive-ratio ranges between $[1,3), [3,5), [5,7), [7,9), [9,11]$. For each group, the average arrival rate and average ratio of demand to sojourn time σ/τ for jobs in each class are shown in (a) and (b), and the number of instances (days) in each class are shown in (c).**

(a) Normalized cost for varying unit penalty of unmet demand.
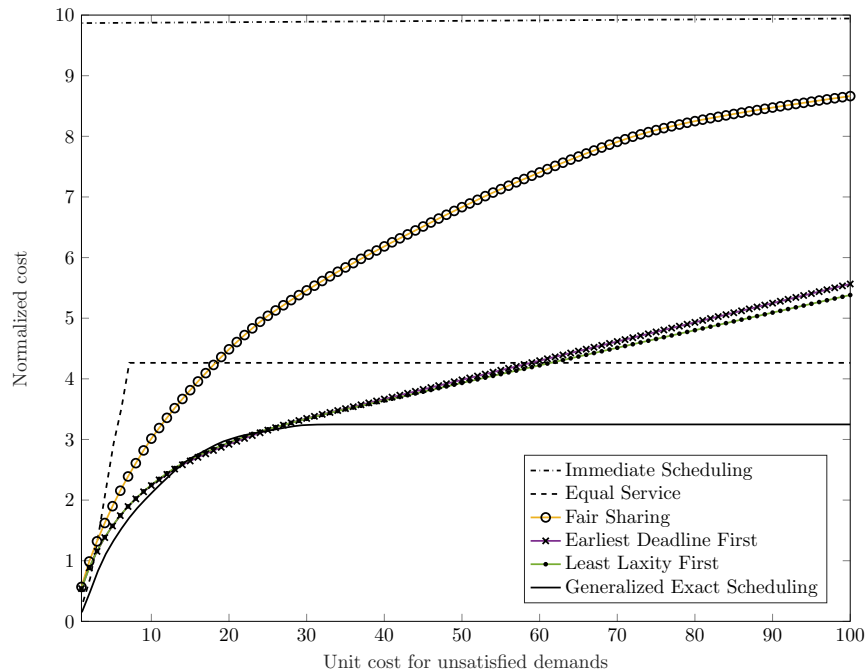
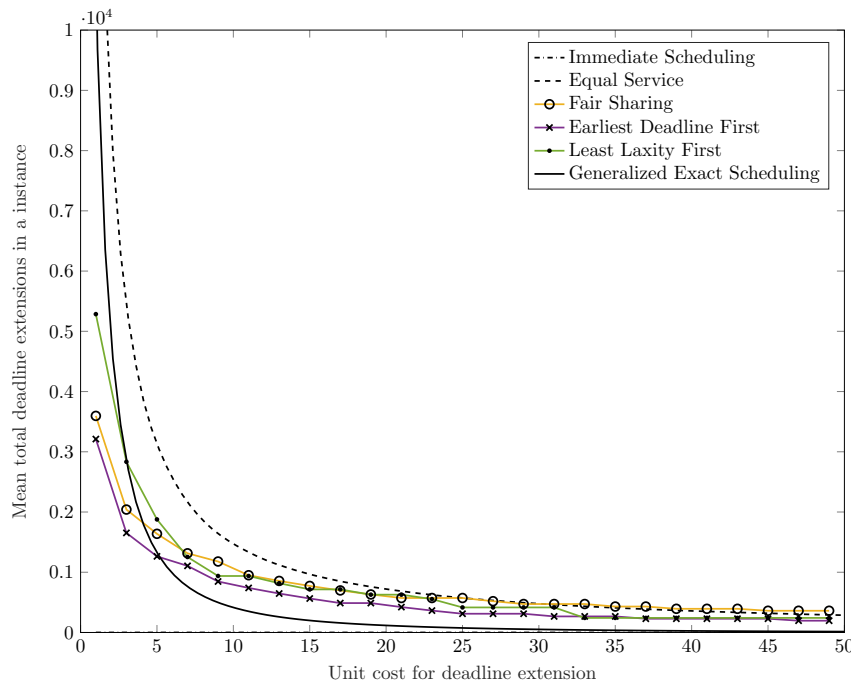(b) Average amount of unsatisfied demands per instance for varying unit penalty of unmet demand.
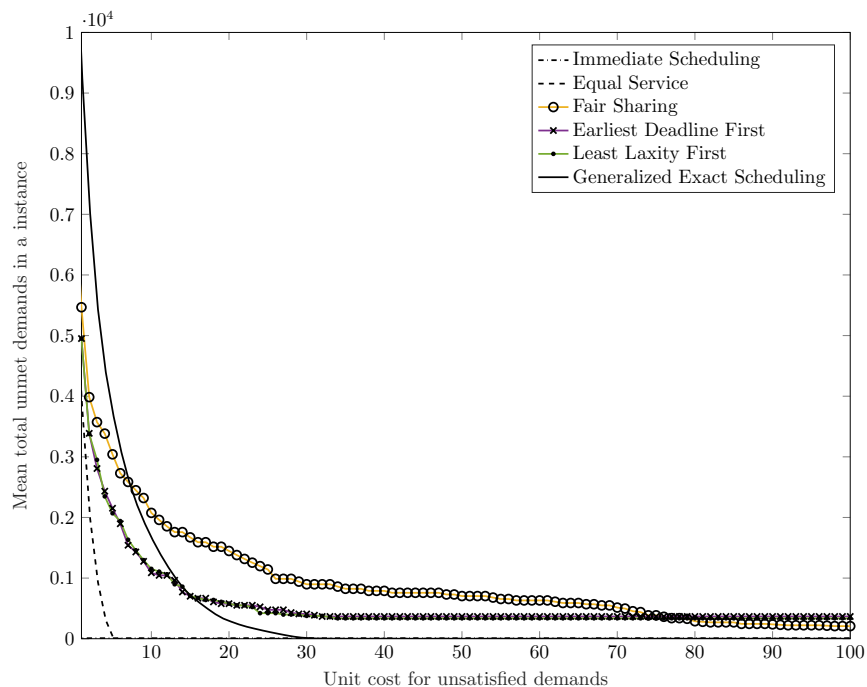
Figure 7   **Generalized Exact Scheduling compared to existing algorithms in the case of soft demand constraints. For varying unit penalty δ, the empirical costs in all instances are shown. The top plot (a) compares the average empirical costs of all instances for varying values unit penalty δ. The cost is normalized by the cost of Offline Optimal (for strict service requirements).[14] The bottom plot (b) shows the average amount of unmet demands in one instance for varying δ. The policy parameters $c_{\mathsf{ES}}, p_{\mathsf{FS}}, p_{\mathsf{EDF}}$, and p_{LLF} used in Equal Service, Fair Sharing, Earliest Deadline First, and Least Laxity First are set to be the optimal offline values that minimize the average empirical costs.[12]**

(a) Normalized cost for varying unit penalty of deadline extension.
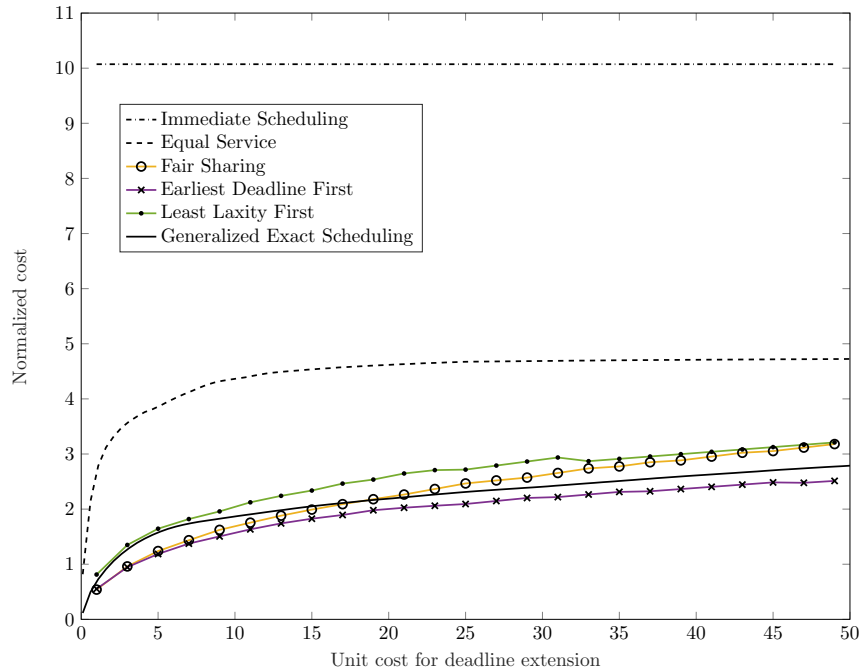
(b) Average amount of deadline extensions per instance for varying unit penalty of deadline extension.

Figure 8 **Generalized Exact Scheduling compared to existing algorithms in the case of soft deadline constraints. For varying unit penalty of unmet demand ϵ, the empirical costs in all instances are shown. The top plot (a) compares the average empirical costs of all instances for varying values unit penalty ϵ. The cost is normalized by the cost of Offline Optimal (for strict service requirements).[14] The bottom plot (b) shows the average amount of deadline extension in one instance for varying ϵ. The policy parameters c_{ES}, p_{FS}, p_{EDF}, and p_{LLF} used in Equal Service, Fair Sharing, Earliest Deadline First, and Least Laxity First are set to be the optimal offline values that minimize the average empirical costs.[12]**

less coordination in scheduling different jobs, which in turn reduces the advantages of being able to use centralized information. Moreover, when the deadline is tight, the service requirements do not allow much flexibility in varying the service rate over time, and the offline (centralized) algorithm may not be able to exploit the future arrival information to its full advantage.

Performance in soft demand constraints. When the demands do not need to be strictly satisfied, the scheduler can exploit this flexibility to reduce the overall cost by balancing the service capacity variance and the penalties for unsatisfied demands. In this setting, the behavior of distributed algorithms (Generalized Exact Scheduling, Immediate Scheduling, Equal Service) and centralized algorithms (Earliest Deadline First, Least Laxity First, Fair Sharing) is compared in Figure 7.[14] As the unit penalty for unmet demands δ grows, all algorithms inevitably suffer from increased costs as well. However, the cost of Generalized Exact Scheduling plateaus at relatively small δ, which results in a lower cost compared with other centralized and distributed algorithms (Figure 7a). This quick plateau is achieved by a highly adaptive reduction in the total amount of unsatisfied demands. For small unit penalty δ, Generalized Exact Scheduling is among the algorithms with the largest amount of unmet demand to exploit the flexibility in being able to miss some demands. For large unit penalty δ, it has the smallest amount of unmet demand in order to minimize its high penalty associated with not meeting demands (Figure 7b). This dynamic and optimal adjustment is obtained as the solution of the optimization problem (7), which balances the service capacity variance and unmet demand. So the design process to achieve this behavior does not require tedious manual adjustments.

Performance in soft deadline constraints. Similarly, when job deadlines do not need to be strictly enforced, the scheduler can exploit this flexibility to reduce the overall cost by balancing the service capacity variance and the penalties for deadline extensions. In this setting, the behavior of a few distributed algorithms and centralized algorithms is compared in Figure 8.[14] Similar to the setting of soft demand, Generalized Exact Scheduling achieves a lower cost than other distributed algorithms, and it also has a comparable performance with the centralized algorithms when the unit penalty for unsatisfied deadline ϵ is large (Figure 8a). Such performance is achieved by drastically reducing the total amount of unsatisfied deadlines as ϵ increases to avoid the high penalty associated with deadline extension (Figure 8b). As $\epsilon \to \infty$, it becomes increasingly costly to miss a job deadline, so the mean total unmet deadlines should converge to zero. This property can be observed in Generalized Exact Scheduling, and the optimal adjustment is obtained as the solution of the optimization problem (8), which systematically balances the service capacity variance and deadline extension.

4.1.4. Dealing with demand and deadline uncertainties Most algorithms discussed in this section require knowledge about the demands and deadlines of all jobs. This condition is valid for certain applications [6, 14, 15, 33]. For example, in the electric vehicle charging testbed [15], the system receives user input about the energy demand and departure time of each vehicle. On the other hand, there are other situations where the information on service requirements (demands and/or deadlines) can be missing for all, or a subset of, jobs.

Recall from Section 3 that the optimal distributed policy is to have a flat and low service rate in the service rate trajectory. This intuition motivates us to consider a mixture of Exact Scheduling and Equal Service: serve according to Exact Scheduling if the demands and deadlines are known; otherwise, make the best guess about a good service rate and apply that rate to all jobs with unknown demand and/or deadlines. This policy reduces to Exact Scheduling if the service requirements for all jobs are known, whereas it reduces to Equal Service when the service requirements for all jobs are unknown. With a slight abuse of notation, we denote this extension for cases of potentially unknown service requirements as Generalized Exact Scheduling as well.

---

[14] In Figure 7-8, we use "normalized cost" instead of empirical competitive-ratio. This is because, unlike the case of empirical competitive-ratio, the cost under soft service requirements is compared here with that of offline optimal for hard service requirements.

Now we look into how much system performance may degrade in $\text{Var}(P)$ if the demands and deadlines are unknown. Let s_k be a binary random variable taking the value of 1 if the system has access to the demand and deadline of job k and 0 otherwise. We assume that the probability of $s = i \in \{1, 0\}$ is $p(s = i)$, and s is independent with a and (σ, τ). Following the argument of (18)-(19), we have

$$
\begin{aligned}
\text{Var}(P) =& \mathbb{E}\left[\int_0^\tau v(\sigma, \tau, y)^2 dy\right] \\
=& p(s=1)\mathbb{E}\left[\frac{\sigma^2}{\tau}\Big|s=1\right] + p(s=0)\mathbb{E}\left[\int_0^\tau v(\sigma, \tau, y)^2 dy\Big|s=0\right] \quad (34) \\
=& p(s=1)\mathbb{E}\left[\frac{\sigma^2}{\tau}\Big|s=1\right] + p(s=0)\mathbb{E}\left[\frac{\sigma^2}{\tau}\Big|s=0\right] \\
& + p(s=0)\mathbb{E}\left[\int_0^\tau v(\sigma, \tau, y)^2 dy\Big|s=0\right] - p(s=0)\mathbb{E}\left[\frac{\sigma^2}{\tau}\Big|s=0\right] \quad (35) \\
=& \mathbb{E}\left[\frac{\sigma^2}{\tau}\right] + p(s=0)\left\{\mathbb{E}\left[\int_0^\tau v(\sigma, \tau, y)^2 dy\Big|s=0\right] - \mathbb{E}\left[\frac{\sigma^2}{\tau}\Big|s=0\right]\right\} \\
=& q_{ES} + p(s=0)\left\{\mathbb{E}\left[\int_0^\tau v(\sigma, \tau, y)^2 dy - (\sigma^2/\tau)\Big|s=0\right]\right\}, \quad (36)
\end{aligned}
$$

where q_{ES} is the optimal cost of Exact Scheduling. Equalities (34) and (35) use the Law of Total Expectation. From (36), the performance degradation due to unknown demands and deadlines is computed to be[15]

$$
\begin{cases}
p(s=0)\mathbb{E}\left[{c'_{ES}}^2 \min\{\tau, \sigma/c'_{ES}\} - (\sigma^2/\tau)\right] & \text{in case of soft demand} \\
p(s=0)\mathbb{E}\left[c''_{ES}\sigma - (\sigma^2/\tau)\right] & \text{in case of soft deadline}
\end{cases}. \quad (37)
$$

For the instances in the testbed and the synthetic data (from Section 4.1.1), the performance degradation is estimated to be $15 \sim 40\%$ of the cost of Exact Scheduling multiplied by the ratio of jobs with unknown service requirements (the overall cost is $100 + 15p(s=0) \sim 100 + 40p(s=0)\%$ of that of Exact Scheduling). This estimation is obtained from the performance difference between Exact Scheduling and Equal Service in Figure 5 by realizing that the value in (37) is upper-bounded by that of Equal Service for strict demand and deadlines requirement.

4.2. Theoretical analysis

In this section, we compare the performance of online distributed policies and that of online centralized policies. The design problem of centralized scheduler is typically formulated as a Markov decision process, whose optimal solution can only be approximated or computed numerically. To obtain analytic bounds, we formulate it as a constrained functional optimization problem instead. Recall from Section 4.1.2 that a centralized scheduling policy has the form c in (29). It can use all available information of jobs arriving prior to time t in decision making. The minimum-variance centralized policy can then be obtained as the solution of the following constrained functional optimization problem

$$
\underset{u:(1)(2)(3)(29)}{\text{minimize}} \quad \text{Var}(P).
$$

where the optimization variable is the scheduling policy of the form (29).

---

[15] No formula is given for the case of strict demand and deadline because demand and deadline satisfaction cannot be guaranteed without the information of demands and deadlines.

In order to bound the performance degradation from centralized to distributed algorithms, we first obtain the performance limits for centralized algorithms. Let $X(t)$ be the total remaining demands of jobs arriving before t. Let D be a value that satisfies

$$\text{Var}(X) \leq D \tag{38}$$

where $\text{Var}(X)$ is the stationary variance of $X(t)$.

Lemma 3. *Under any centralized policy of the form* (29), *the stationary variance of $P(t)$ is lower-bounded by*

$$\text{Var}(P) \geq \frac{1}{4D}\Lambda^2\mathbb{E}[\sigma^2]^2.$$

Corollary 2. *Let* $\text{Var}(P)$ *be the stationary variance of $P(t)$ attained by Exact Scheduling* (13). *Let* $\text{Var}(P^\dagger)$ *be the optimal performance among the centralized scheduling policies of the form* (29) *that satisfy the rate, demand, and deadline constraints* (1)–(3). *Then, the following inequality holds:*

$$\text{Var}(P) \leq \frac{4\mathbb{E}\left[\sigma^2/\tau\right]\left(\mathbb{E}[\tau\sigma^2] + \Lambda\mathbb{E}\left[\tau\sigma\right]^2\right)}{\mathbb{E}[\sigma^2]^2}\text{Var}(P^\dagger). \tag{39}$$

The proof of Corollary 2 is given in Appendix G. Corollary 2 bounds the ratio of stationary variance achievable by the optimal distributed algorithm to that achievable by any centralized algorithms. Here, both the optimal distributed algorithm and the optimal centralized algorithm are subject to strict constraints on demands (2) and deadlines (3). To evaluate this bound, consider a special case where both σ and τ are deterministic and $\sigma = a\tau$ for some scalar $a > 1$. Then bound (39) reduces to $\text{Var}(P) \leq 4(1+\Lambda\tau)\text{Var}(P^\dagger)$. This formula suggests that Exact Scheduling becomes more competitive to Centralized Optimal algorithms when arrival rate is small. This observation is consistent with the observation that large performance difference between Exact Scheduling and Offline Optimal mostly happens at instances with large arrival rate in the testbed (Figure 6a). As $\Lambda\tau \to 0$, the bound suggests that the cost of Exact Scheduling remains within approximately 4 times of the cost of optimal centralized algorithm. Recall from Figure 5a that the cost of Exact Scheduling performs approximately 3 times of that of Offline Optimal, and 1.2 times of that of Online Optimization MPC. This data suggests Exact Scheduling may perform much better than this performance lower-bound suggests. The pessimistic estimate of bound (39) may be due to the fact that the proof of Corollary 2 uses a loose bound for (38) (see (160)–(161) in the Appendix G). Alternatively, a tighter bound can be obtained as

$$\text{Var}(P) \leq \frac{4D\mathbb{E}\left[\sigma^2/\tau\right]}{\mathbb{E}[\sigma^2]^2}\text{Var}(P^\dagger).$$

where an estimate of D can be computed numerically given the arrival distribution.

4.2.1. Proof of Lemma 3 In this section, we present the proof of Lemma 3. Let the stationary variance of $X(t)$ be bounded as in (38). We consider the following problem:

$$\mathcal{Q}_{\text{on}} = \underset{c:(29)(38)}{\text{minimize}} \ \lim_{T\to\infty}\frac{1}{T}\int_0^T \text{Var}(P(t))dt,$$

where the optimization is taken over all centralized policies of the form (29) satisfying (38). The Lagrangian of $\mathcal{Q}_{\text{on}}$ is

$$L(c;\gamma) = \lim_{T\to\infty}\frac{1}{T}\int_0^T \text{Var}(P(t)) + \gamma(\text{Var}(X(t)) - D)dt,$$

where $\gamma \geq 0$ is the Lagrangian multiplier associated with the constraint (38). Observe that

$$\inf_{c:(29)} L(c; \gamma) \leq \mathcal{Q}_{\text{on}} \leq \text{Var}(P), \tag{40}$$

where $\text{Var}(P)$ is the stationary service capacity variance of any policy. Then, we can derive a lower bound of $\text{Var}(P)$ via solving $\inf_{c:(29)} L(c; \gamma)$ as follows.

Lemma 4. *Let $\bar{X}$ and $\bar{P}$ be defined as the stationary mean of $X(t)$ and $P(t)$, respectively. The infimum in $\inf_{c:(29)} L(c; r)$ is attained when $P(t)$ is set to be*

$$P(t) = \sqrt{\gamma}(X(t) - \bar{X}) + \bar{P} \tag{41}$$

at all time t, and the infimum value is given by

$$\inf_{c:(29)} L(c; \gamma) = \sqrt{\gamma} \Lambda \mathbb{E}[\sigma^2] - \gamma D \tag{42}$$

$$= \sqrt{\gamma} \Lambda \mathbb{E}[\sigma^2] - \gamma \text{Var}(X). \tag{43}$$

Lemma 4 is proven in Appendix F. From (41), the optimal solution of $\inf_{c:(29)} L(c; \gamma)$ satisfies

$$\text{Var}(P) + \gamma \text{Var}(X) = 2\gamma \text{Var}(X). \tag{44}$$

Combining (42) and (44) leads to

$$\text{Var}(X) = \frac{1}{2\sqrt{\gamma}} \Lambda \mathbb{E}[\sigma^2]. \tag{45}$$

Since $X(t)$ also satisfies the constraint (38), the Lagrangian multiplier γ is lower-bounded by

$$\frac{1}{2D} \Lambda \mathbb{E}[\sigma^2] \leq \sqrt{\gamma}. \tag{46}$$

Therefore, we obtain

$$\text{Var}(P) \geq \inf_{c:(29)} L(c; \gamma) \tag{47}$$

$$= \sqrt{\gamma} \Lambda \mathbb{E}[\sigma^2] - \gamma \text{Var}(X) \tag{48}$$

$$= \frac{\sqrt{\gamma}}{2} \Lambda \mathbb{E}[\sigma^2] \tag{49}$$

$$\geq \frac{1}{4D} \Lambda^2 \mathbb{E}[\sigma^2]^2. \tag{50}$$

where (47) is due to (40); (48) is due to (43); (49) is due to (45); and (50) is due to (46).

5. Balancing predictability and stability under non-stationary job arrivals

Building upon the results of stationary job arrivals, we consider a more general setting with non-stationary job arrivals in this section. The non-stationary setting is particularly appealing for practical applications since dynamic capacity management is most crucial when the workload is not stationary.

In contrast to the stationary setting, there exists a tradeoff between maximizing the stability and predictability of the service capacity in the non-stationary setting. We characterize this tradeoff and introduce a Pareto-optimal distributed algorithm that balances stability and predictability. Below, we first formally define the notion of Pareto-optimality, which recovers maximum predictability and maximum stability as two special cases (Section 5.1). Then, at one extreme case of maximizing predictability, we show that Generalized Exact Scheduling is the optimal algorithm (Section 5.2). In the other extreme case of maximizing stability, we characterize the optimal algorithm and notice an interesting connection to the well-known YDS algorithm [14], which is optimal in a related, deterministic worst-case setting (Section 5.3). Generalizing the two extreme cases, we describe the Pareto-optimal algorithm that balances predictability and stability (Section 5.4).

5.1. Problem formulation

In this section, we relax our previous stationary assumptions on the arrival process. We assume that the arrival distribution is a non-stationary independently marked Poisson process with the intensity function $\tilde{\Lambda}(a)$ and a mark joint density measure $f_a(\sigma, \tau) g_a(\delta) h_a(\epsilon)$, and let $\Lambda(a, \sigma, \tau) = \tilde{\Lambda}(a) f_a(\sigma, \tau) g_a(\delta) h_a(\epsilon)$ (see Section 2).

We assume that the time horizon is chosen such that the following conditions hold:

$$\Lambda(a, \sigma, \tau) = 0, \qquad\qquad\qquad \forall a < 0 \qquad\qquad (51)$$

$$T \geq a + \tau, \qquad\qquad \forall(a, \sigma, \tau) \text{ with } \Lambda(a, \sigma, \tau) > 0. \qquad (52)$$

Condition (51) assumes jobs to arrive after time $t = 0$, and condition (52) sets the time horizon T to be long enough so that service can be provided during the sojourn time to all jobs. We consider the following three types of policies:

$$r_k(t) = u(a_k, x_k(t), y_k(t)) \geq 0 \quad k \in \mathcal{V} \qquad\qquad (53)$$

$$r_k(t) = \bar{u}(a_k, x_k(t), y_k(t), \delta_k, \epsilon_k) \geq 0 \quad k \in \mathcal{V} \qquad (54)$$

$$r_k(t) = v(a_k, \sigma_k, \tau_k, y_k(t)) \geq 0 \quad k \in \mathcal{V}, \qquad\qquad (55)$$

Here, to better account for changes in the arrival distribution, we allow the scheduling policies u, $\bar{u}$, v to be a function of the arrival time.[16] They are also online and distributed in the sense that the service rate of each job is determined using only the information of the same job but not other jobs. Policies of the form (53) is a relaxation of policies of the form (4) in the sense that any policy $u(x_k(t), y_k(t))$ in (4) can be represented by some $u(a_k, x_k(t), y_k(t))$ in (53).

We seek to design policies with enhanced predictability, quantified by the variance of the service capacity, and stability, quantified by the mean square of the service capacity. In the settings when predictability of the service capacity is the only concern, we consider the optimization problem

$$\underset{u:(1)(2)(3)(53)}{\text{minimize}} \quad \lim_{T \to \infty} \frac{1}{T} \int_0^T \text{Var}(P(t)) dt \qquad\qquad (56)$$

for the case of strict service requirements and the optimization problem

$$\underset{\bar{u}:(1)(54)}{\text{minimize}} \quad \lim_{T \to \infty} \frac{1}{T} \int_0^T \Big(\text{Var}(P(t)) + \mathbb{E}[U(t)] + \mathbb{E}[W(t)] \Big) dt \qquad (57)$$

for the case of soft service requirements. In the settings when stability of the service capacity is the only concern, we consider relaxing the service rate constraint as

$$r_k(t) \geq 0, \qquad\qquad k \in \mathcal{V}, t \in [0, T] \qquad\qquad (58)$$

and solve the optimization problem

$$\underset{v:(2)(3)(55)(58)}{\text{minimize}} \quad \lim_{T \to \infty} \frac{1}{T} \int_0^T \mathbb{E}[P(t)]^2 dt. \qquad\qquad (59)$$

In the settings involving both predictability and stability of the service capacity, we consider the optimization problem

$$\underset{v:(1)(2)(3)(55)}{\text{minimize}} \quad \lim_{T \to \infty} \frac{1}{T} \int_0^T \alpha \mathbb{E}[P(t)]^2 + \beta \text{Var}(P(t)) dt \qquad (60)$$

for the case of strict service requirements. Note that this problem gives Pareto-optimal policy that balances predictability and stability.

More optimization problems can be formulated by combining the terms in (56)–(59). Although such optimization problems are beyond the scope of this paper, our techniques can be used to analyze such problems as well.

---

[16] In other words, two jobs with identical pairs of (σ_k, τ_k) can be served differently based on their arrival times.

5.2. Maximizing predictability

In this section, we consider the optimization problem of maximizing the predictability in service capacity (*i.e.* minimizing the service capacity variance) in the settings of strict service requirements and soft service requirements. The problem allows us to systematically balance the service quality and service capacity variance and also admits insightful closed-form solutions. Recall from Section 3.2 that $\mathrm{Var}(P(t))$ is minimized at a flat service rate because having peaks and fluctuations in the service rate within a job's sojourn time amplifies the uncertainties of the future arrivals to cause large $\mathrm{Var}(P(t))$. This intuition holds beyond stationary arrivals, and so does for the optimal algorithm for non-stationary arrival distribution.

Theorem 5. *The optimal solution of* (56) *is Exact Scheduling, defined by*

$$u(a,x,y) = \begin{cases} \dfrac{x}{y} & x > 0, y > 0 \\ 0 & otherwise \end{cases}.$$

Furthermore, Generalized Exact Scheduling is also optimal under soft demand and deadline constraints, despite the non-stationary arrival distribution.

Corollary 3. *The optimal solution of* (57) *is*

$$\bar{u}(a,x,y,\delta,\epsilon) = \begin{cases} \dfrac{x}{y} & if\ y > 0\ and\ \dfrac{x}{y} \leq \min\left\{\dfrac{\delta}{2}, \sqrt{\epsilon}\right\} \\ \dfrac{\delta}{2} & if\ y > 0\ and\ \dfrac{x}{y} > \dfrac{\delta}{2}\ and\ \dfrac{\delta}{2} \leq \sqrt{\epsilon} \\ \sqrt{\epsilon}\,\mathbf{1}\{x > 0\} & otherwise \end{cases},$$

Analogously to its stationary-case counterpart in Section 3.5, unit costs for unmet demands and deadlines (δ_k, ϵ_k) determines the tradeoffs between reducing service capacity variance versus allowing unmet demands or deadline extension as Figure 4. The policy operates in three regimes:

- *High penalties regime.* Both demands and deadlines are satisfied. The service rates are identical to the ones produced by Exact Scheduling (13).
- *Low demand penalty regime.* All deadlines are strictly enforced with potentially unsatisfied demands. The service rates are identical to the ones produced by policy (21).
- *Low deadline penalty regime.* Demands are strictly satisfied with potential deadline extensions. The service rates are identical to the ones produced by policy (23).

Corollary 3 reduces to Theorem 5 as the unit costs for unmet demands and deadlines (δ_k, ϵ_k) approach infinity. Corollary 3 also recovers the results from section 3 as the special cases: When the job arrival distribution is stationary, it becomes Corollary 1.

To prove Theorem 5, we take analogous steps to Theorem 1. We consider the optimization problem that relaxes the form of the scheduling policy from (53) to (55):

$$\underset{v:(1)(2)(3)(55)}{\text{minimize}} \int_0^T \mathrm{Var}(P(t))dt, \tag{61}$$

where the time horizon T is assumed to be finite. Compared with the stationary setting, obtaining a closed-form solution of (61) requires additional treatment to account for the non-stationarity in arrival distribution.

Lemma 5. *Exact Scheduling $v(a,\sigma,\tau,y) = (\sigma/\tau)\mathbf{1}\{y > 0\}$ is the optimal solution of* (61).

The proof of Lemma 5 uses the following lemma.

Lemma 6. *The mean and variance of $P(t)$ under the policy* (55) *are given by*

$$\mathbb{E}[P(t)] = \int_{(\sigma,\tau)\in S} \int_0^\tau v(t+y-\tau,\sigma,\tau,y)\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau \tag{62}$$

$$\mathrm{Var}(P(t)) = \int_{(\sigma,\tau)\in S} \int_0^\tau v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau. \tag{63}$$

Lemma 6 can be proved following similar steps to the proof of Lemma 2 (see Appendix A for more detail). Now we are ready to prove Lemma 5.

Proof of Lemma 5. From Lemma 6, the objective function of (61) satisfies

$$\int_0^T \mathrm{Var}(P(t))dt = \int_{t=0}^T \int_{(\sigma,\tau)\in S} \int_{y=0}^\tau v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau dt$$

$$= \int_{(\sigma,\tau)\in S} \left\{ \int_{y=0}^\tau \int_{t=0}^T v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dtdy \right\} d\sigma d\tau.$$

Moreover, the constraints of (61) can be rewritten into

$$\int_{y=0}^\tau v(a,\sigma,\tau,y)dy = \sigma \qquad\qquad a\in\mathcal{T}, (\sigma,\tau)\in S \tag{64}$$

$$0 \le v(a,\sigma,\tau,y) \le 1 \qquad\qquad a\in\mathcal{T}, (\sigma,\tau)\in S, y\in[0,\tau] \tag{65}$$

For any $(\sigma,\tau)\in S$, the optimal solution of (61) is attained at the minimum of the following optimization problem:

$$\underset{v:(64)(65)}{\text{minimize}} \int_{y=0}^\tau \int_{t=0}^T v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dtdy \tag{66}$$

From integration by substitution, the objection function of (66) satisfies

$$\int_{y=0}^\tau \int_{t=0}^T v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dtdy = \int_{y=0}^\tau \int_{a=y-\tau}^{T+y-\tau} v(a,\sigma,\tau,y)^2\Lambda(a,\sigma,\tau)dady$$

$$= \int_{y=0}^\tau \int_{a=0}^T v(a,\sigma,\tau,y)^2\Lambda(a,\sigma,\tau)dady,$$

where the last equality is due to the assumption that the time-horizon is chosen to be large enough to satisfy $\Lambda(a,\sigma,\tau)=0$ for any $a\in[y-\tau,T+y-\tau]\setminus[0,T]$. The Lagrangian of (66) is

$$L(v;\mu,\nu) = \int_{y=0}^\tau \int_{a=y-\tau}^{T+y-\tau} v(a,\sigma,\tau,y)^2\Lambda(a,\sigma,\tau)dady - \int_{a=0}^T \mu_{\sigma,\tau}(a)\int_{y=0}^\tau v(a,\sigma,\tau,y)dyda$$

$$+ \int_{a=0}^T \int_{y=0}^\tau (\bar{\nu}_{\sigma,\tau}(a,y) - \underline{\nu}_{\sigma,\tau}(a,y))v(a,\sigma,\tau,y)dyda,$$

where $\mu_{\sigma,\tau}(a)$ is the Lagrange multiplier associated with constraint (64); $\underline{\nu}_{\sigma,\tau}(a,y)\ge 0$ is the Lagrange multiplier associated with the constraint $v(a,\sigma,\tau,y)\ge 0$, and $\bar{\nu}_{\sigma,\tau}(a,y)\ge 0$ is the Lagrange multiplier associated with the constraint $v(a,\sigma,\tau,y)\le 1$. A necessary condition for v^* to be the optimal scheduling policy is that $L(v;\mu,\nu)$ is stationary at $v=v^*$. After some manipulation, the stationary condition can be computed as follows: for any (a,σ,τ) with $\Lambda(a,\sigma,\tau)>0$,

$$v^*(a,\sigma,\tau,y) = \frac{\mu_{\sigma,\tau}(a) + \underline{\nu}_{\sigma,\tau}(a,y) - \bar{\nu}_{\sigma,\tau}(a,y)}{2\Lambda(a,\sigma,\tau)}, \tag{67}$$

for $y \in [0, \tau]$. In addition, we have

$$\forall y \text{ s.t. } v(a, \sigma, \tau, y) < 1, \qquad \qquad \bar{\nu}_{\sigma, \tau}(a, y) = 0, \qquad \qquad (68)$$

$$\forall y \text{ s.t. } v(a, \sigma, \tau, y) > 0, \qquad \qquad \underline{\nu}_{\sigma, \tau}(a, y) = 0. \qquad \qquad (69)$$

Next, we fix (a, σ, τ), assuming $\Lambda(a, \sigma, \tau) > 0$, and use proof by contradiction to show that the optimal service rate $v^*(a, \sigma, \tau, y)$ remains constant for throughout the sojourn time of a job, *i.e.*

$$v^*(a, \sigma, \tau, y_1) = v^*(a, \sigma, \tau, y_2) \qquad \qquad \forall y_1, y_2 \in [0, \tau]. \qquad \qquad (70)$$

Suppose not, *i.e.*

$$\exists y_1 \neq y_2 \in [0, \tau], \qquad \qquad v^*(a, \sigma, \tau, y_1) \neq v^*(a, \sigma, \tau, y_2),$$

then v^* must satisfy one of the following cases:
Case 1

$$1 > v^*(a, \sigma, \tau, y_1) > v^*(a, \sigma, \tau, y_2) > 0, \qquad \qquad (71)$$

Case 2

$$1 > v^*(a, \sigma, \tau, y_1) > v^*(a, \sigma, \tau, y_2) = 0, \qquad \qquad (72)$$

Case 3

$$1 = v^*(a, \sigma, \tau, y_1) > v^*(a, \sigma, \tau, y_2) = 0, \qquad \qquad (73)$$

Case 4

$$1 = v^*(a, \sigma, \tau, y_1) > v^*(a, \sigma, \tau, y_2) > 0. \qquad \qquad (74)$$

Suppose Case 1 holds, then we have

$$v^*(a, \sigma, \tau, y_1) = \frac{\mu_{\sigma, \tau}(a)}{2\Lambda(a, \sigma, \tau)} = v^*(a, \sigma, \tau, y_2), \qquad \qquad (75)$$

from (67), (68), (69), and (71). However, (75) contradicts (71).

Suppose Case 2 holds, then we have

$$v^*(a, \sigma, \tau, y_1) = \frac{\mu_{\sigma, \tau}(a)}{2\Lambda(a, \sigma, \tau)} > 0 \qquad \qquad (76)$$

$$v^*(a, \sigma, \tau, y_2) = \frac{\mu_{\sigma, \tau}(a) + \underline{\nu}_{\sigma, \tau}(a, y_2)}{2\Lambda(a, \sigma, \tau)} = 0 \qquad \qquad (77)$$

from (67), (68), (69), and (72). Combining (76) and (77) leads to

$$0 = \frac{\mu_{\sigma, \tau}(a) + \underline{\nu}_{\sigma, \tau}(a, y_2)}{2\Lambda(a, \sigma, \tau)} < \frac{\mu_{\sigma, \tau}(a)}{2\Lambda(a, \sigma, \tau)}. \qquad \qquad (78)$$

However, $\underline{\nu}_{\sigma, \tau}(a, y_2) \geq 0$ implies

$$\frac{\mu_{\sigma, \tau}(a) + \underline{\nu}_{\sigma, \tau}(a, y_2)}{2\Lambda(a, \sigma, \tau)} \geq \frac{\mu_{\sigma, \tau}(a)}{2\Lambda(a, \sigma, \tau)},$$

which contradicts (78).

Suppose Case 3 holds, then we have

$$v^*(a,\sigma,\tau,y_1) = \frac{\mu_{\sigma,\tau}(a) - \bar{\nu}_{\sigma,\tau}(a,y_1)}{2\Lambda(a,\sigma,\tau)} = 1 \tag{79}$$

$$v^*(a,\sigma,\tau,y_2) = \frac{\mu_{\sigma,\tau}(a) + \underline{\nu}_{\sigma,\tau}(a,y_2)}{2\Lambda(a,\sigma,\tau)} = 0, \tag{80}$$

from (67), (68), (69), and (73). From (79) and (80), we have

$$1 = \frac{\mu_{\sigma,\tau}(a) - \bar{\nu}_{\sigma,\tau}(a,y_1)}{2\Lambda(a,\sigma,\tau)} > \frac{\mu_{\sigma,\tau}(a) + \underline{\nu}_{\sigma,\tau}(a,y_2)}{2\Lambda(a,\sigma,\tau)} = 0. \tag{81}$$

Combining $\bar{\nu}_{\sigma,\tau}(a,y_1) \geq 0$ and $\underline{\nu}_{\sigma,\tau}(a,y_2) \geq 0$ yields

$$\frac{\mu_{\sigma,\tau}(a) - \bar{\nu}_{\sigma,\tau}(a,y_1)}{2\Lambda(a,\sigma,\tau)} \leq \frac{\mu_{\sigma,\tau}(a) + \underline{\nu}_{\sigma,\tau}(a,y_2)}{2\Lambda(a,\sigma,\tau)},$$

which contradicts (81).

Suppose Case 4 holds, then we have

$$v^*(a,\sigma,\tau,y_1) = \frac{\mu_{\sigma,\tau}(a) - \bar{\nu}_{\sigma,\tau}(a,y_1)}{2\Lambda(a,\sigma,\tau)} = 1 \tag{82}$$

$$v^*(a,\sigma,\tau,y_2) = \frac{\mu_{\sigma,\tau}(a) + \underline{\nu}_{\sigma,\tau}(a,y_2)}{2\Lambda(a,\sigma,\tau)} = 0, \tag{83}$$

from (67), (68), (69), and (74). Combining (82) and (83) lead to

$$1 = \frac{\mu_{\sigma,\tau}(a) - \bar{\nu}_{\sigma,\tau}(a,y_1)}{2\Lambda(a,\sigma,\tau)} > \frac{\mu_{\sigma,\tau}(a) + \underline{\nu}_{\sigma,\tau}(a,y_2)}{2\Lambda(a,\sigma,\tau)} = 0. \tag{84}$$

However, $\bar{\nu}_{\sigma,\tau}(a,y_1) \geq 0$ and $\underline{\nu}_{\sigma,\tau}(a,y_2) \geq 0$ give

$$\frac{\mu_{\sigma,\tau}(a) - \bar{\nu}_{\sigma,\tau}(a,y_1)}{2\Lambda(a,\sigma,\tau)} \leq \frac{\mu_{\sigma,\tau}(a) + \underline{\nu}_{\sigma,\tau}(a,y_2)}{2\Lambda(a,\sigma,\tau)},$$

which contradicts (84).

Since none Cases 1-4 can hold, the optimal service rate remains constant throughout the sojourn time of a job, *i.e.* condition (70) holds. Therefore, the optimal solution of (61) is $v(a,\sigma,\tau,y) = (\sigma/\tau)1\{y > 0\}$, which is Exact Scheduling.   Q.E.D.

Note that the optimal value of Exact Scheduling can be represented by control policy of the form (53), and the optimal value of (56) is lower bounded by that of (61). Therefore, Exact Scheduling is also optimal for (56), yielding Theorem 5.

The optimization problem (9) can also be solved in a similar manner. The optimal solution of (56) is also the pointwise minimum of
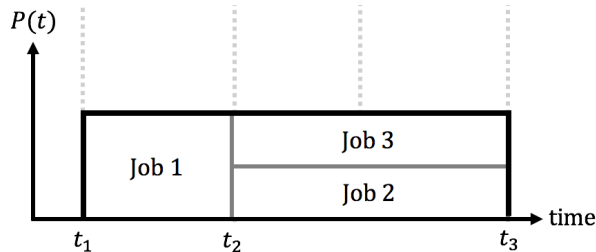
$$\int_{a=0}^{T} \int_{\mathbb{R}_+} \int_{\mathbb{R}_+} \left\{ \int_{y=0}^{\tau} v(a,\sigma,\tau,y)^2 + \delta v(a,\sigma,\tau,y)dy + \epsilon(\hat{\tau}(a,\sigma,\tau) - \tau) \right\} \Lambda(a,\sigma,\tau)f(\delta)f(\epsilon)d\delta d\epsilon da.$$
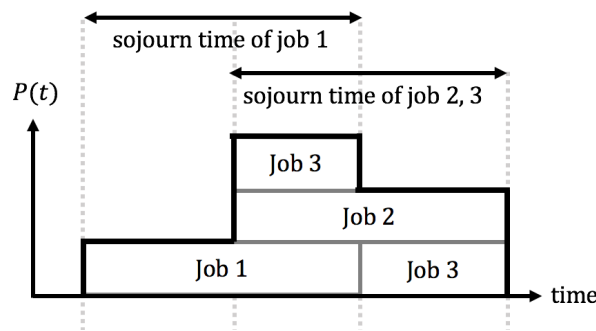
From this observation, we can get Corollary 3 by computing

$$v* = \arg\min_{v} \left\{ \int_{y=0}^{\tau} v(a,\sigma,\tau,y)^2 + \delta v(a,\sigma,\tau,y)dy + \epsilon(\hat{\tau}(a,\sigma,\tau) - \tau) \right\}$$

and converting v^* to take the form (55).

(a) The behavior of Exact Scheduling. Exact Scheduling is likely to incur a substantial cost at a later time.

(b) Ideal behavior. The service rate of job 1 is increased to account for potentially large arrivals in the future.

Figure 9   **This example demonstrates why Exact Scheduling does not maximize stability. This instance has a small arrival rate initially but higher arrival rate at a later time.**

5.3. Maximizing stability

In this section, we consider the optimization problem of maximizing the stability in service capacity (*i.e.* minimizing the service capacity mean square) in the settings of strict service requirements. This problem yields a scheduler that has a striking analogy to the YDS algorithm [14], an optimal scheduler in a deterministic problem.

Recall from Section 3 that achieving stability and predictability are not mutually conflicting goals when it comes to the design of distributed algorithms given a stationary arrival distribution. However, when the arrival process is non-stationary, there is a tradeoff between these two goals, and the minimum-service-capacity-variance algorithm (Exact Scheduling) does not minimize the service capacity mean square. This fact can be easily seen in the example instance of Figure 9. In this instance, the arrival rate increases over time, and Exact Scheduling is likely to incur a substantial cost at a later time (Figure 9a). Meanwhile, an ideal algorithm should account for the increment in future arrivals by serving previous jobs more aggressively than Exact Scheduling (Figure 9).

Formally, the optimal algorithm must satisfy the condition stated below.

Corollary 4. *The optimal solution of* (59) *has the following properties: for each job profile* $(a, \sigma, \tau) \in (\mathcal{T}, S)$,
  (i) $\mathbb{E}[P(h)]$ *takes a constant value for any time* h *at which* $v(a, \sigma, \tau, a + \tau - h) > 0$.
  (ii) *If* $v(a, \sigma, \tau, a + \tau - h) > 0$ *for some* h *and* $v(a, \sigma, \tau, a + \tau - h') = 0$ *for some* h', *then* $\mathbb{E}[P(h')] \geq \mathbb{E}[P(h)]$.

Corollary 4 is a special case of Theorem 6, which is presented in Section 5.4. Corollary 4 states that a job receives a non-zero service rate only during the period at which the service capacity is expected to be low. Specifically, if the service capacity is expected to be lower at h than h', no service should be provided at h' without first providing service at h (without exploiting the expected low service capacity level at h). This desired property is formalized into conditions (i)

and (ii). Condition (i) flattens out the service capacity for times at which the service rate is non-zero. The flat service capacity is desirable because fluctuations of $\mathbb{E}[P(h)]$ during these intervals compromise stability. Condition (ii) picks the period of lowest expected service capacity to serve.

The impact of conditions (i) and (ii) are illustrated intuitively in Figure 9b. Condition (i) forces $\mathbb{E}[P(t)]$ to be constant during the time intervals $[t_1, t_2]$ and $[t_2, t_3]$. Condition (ii) constrains $\mathbb{E}[P(h')] \geq \mathbb{E}[P(h)]$ for any $h \in [t_1, t_2), h' \in [t_2, t_3)$ so that, during interval $[t_1, t_2]$, job 1 is not served beyond an extent that makes $\mathbb{E}[P(h)]$ in $[t_1, t_2]$ higher than $\mathbb{E}[P(h')]$ in $[t_2, t_3]$. Consequently, the resulting service capacity has less fluctuations.

The above properties are commonly observed in 'Valley Filling' or 'Water Filling' algorithms. Algorithms of this type were previously proposed for budget allocation problems such as CPU scheduling [14], temperature and energy control [6], electric vehicle charging in deterministic settings [33, 47], Parallel Gaussian Channels [48, Chapter 9], and optimal packet scheduling [49]. Furthermore, the optimal policy in Corollary 4 also has an interesting similarity to the YDS algorithm [6, 14]. Specifically, the YDS algorithm finds the scheduling rate $r_k(t), k \in \mathcal{V}, t \in [0, T]$ that optimizes

$$\underset{r:(2)(3)(58)}{\text{minimize}} \quad \frac{1}{T} \sum_{t=0}^{T} P(t)^{\alpha} \tag{85}$$

where $\alpha > 1$ is some constant. The optimal solution of (85) satisfies the following conditions: for any job $k \in \mathcal{V}$,

(iii) $P(h)$ takes a constant value at any time h at which $r_k(h) \geq 0$.

(iv) If $r_k(h) > 0$ for some h and $r_k(h') = 0$ for some h', then $P(h') \geq P(h)$.

When we replace $\mathbb{E}[P(t)]$ with $P(t)$ and u with r, conditions (i)–(ii) in Corollary 4 become conditions (iii)–(iv) above. This relationship allows us to adapt the computational tool of the YDS algorithm to find the optimal distributed policy in our setting.

Next, we present a computational method to find the optimal distributed policy that maximizes stability. Let $\mathcal{V}(t_1, t_2) = \{(a, \sigma, \tau) : a \geq t_1, a + \tau \leq t_2, (\sigma, \tau) \in S\}$ be the set of job profiles that have an arrival time after t_1 and a deadline before t_2. We say that jobs with (a, σ, τ) are present in the interval $[t_1, t_2]$ when $(a, \sigma, \tau) \in \mathcal{V}(t_1, t_2)$. Let $w(t_1, t_2)$ denote the expected cumulative demand of jobs present in the interval $[t_1, t_2]$, *i.e.*

$$w(t_1, t_2) = \int_{a \geq t_1} \int_{\tau \leq t_2 - a, (\sigma, \tau) \in S} \sigma \Lambda(a, \sigma, \tau) d\sigma d\tau da.$$

Intuitively, $w(t_1, t_2)$ is the minimum expected demand that must be supplied during a time interval $[t_1, t_2]$ to satisfy the demand requirements. We further define the intensity of an interval $[t_1, t_2]$ as

$$I(t_1, t_2) = \frac{w(t_1, t_2)}{t_2 - t_1}.$$

Now, we are ready to present the computational method in Algorithm 1.

This method finds the service rate $v^*(a, \sigma, \tau, y)$ in descending order of the intensity $I(t_1, t_2)$ in which a job present. Specifically, it iterates the following procedures. It first finds

$$[t_1, t_2] = \arg \max_{[t_1, t_2]} I(t_1, t_2), \tag{86}$$

which is a maximum intensity interval (line 4). Then, it computes the scheduling rates that satisfy

$$\mathbb{E}[P(h)] = \frac{I(t_1, t_2)}{t_2 - t_1} \quad h \in [t_1, t_2) \tag{87}$$

$$v^*(a, \sigma, \tau, y) \leq \bar{v}(a, \sigma, \tau, y). \tag{88}$$

Algorithm 1: Computing the optimal distributed policy that maximizes stability

input : $\Lambda(a,\sigma,\tau)$
output: $v^*(a,\sigma,\tau,y)$
1 initialize $\bar{v}(a,\sigma,\tau,y) \leftarrow \infty$;
2 initialize $\hat{\Lambda}(a,\sigma,\tau) \leftarrow \Lambda(a,\sigma,\tau)$;
3 while $\hat{\Lambda}(a,\sigma,\tau) > 0$ *for some* (a,σ,τ) **do**
4   identify the maximum intensity interval $[t_1,t_2]$ by solving (86);
5   compute $v^*(a,\sigma,\tau,y)$ for job profiles in $\mathcal{V}(t_1,t_2)$ s.t. (87) and (88);
6   **for** $(a,\sigma,\tau) \notin \mathcal{V}(t_1,t_2)$ **do**
7   set $\bar{v}(a,\sigma,\tau,y) \leftarrow 0$ for any $a + \tau - y \in [t_1,t_2]$;
8   set $\hat{\Lambda}(a,\sigma,\tau) \leftarrow 0$ for job profiles in $\mathcal{V}(t_1,t_2)$;

---

for job profiles present in the maximum intensity interval (line 5). As jobs not present in $\mathcal{V}(t_1,t_2)$ are assigned zero service capacity during $[t_1,t_2]$, the maximum service rates $\bar{v}(a,\sigma,\tau,y)$ of jobs not present in $\mathcal{V}(t_1,t_2)$ are set to be zero during $[t_1,t_2]$ (line 7). From (88), these jobs will be assigned zero service rate during interval $[t_1,t_2]$ in later loops. Because jobs in $\mathcal{V}(t_1,t_2)$ are already scheduled, before the next iteration, they are removed from the arrival statistics used to compute the maximum intensity interval of the unscheduled jobs by setting $\hat{\Lambda}(a,\sigma,\tau) = 0$ for job profiles in $\mathcal{V}(t_1,t_2)$ (line 8). Using the modified arrival statistics, the algorithm repeats the same process of finding a new maximum intensity interval, computing the service rates of jobs present during this interval, and modifying job statistics.

The following two corollaries state the properties which Algorithm 1 satisfies.

Corollary 5. *The output of Algorithm 1 satisfies the conditions (i) and (ii) stated in Corollary 4.*

Proof. For each job profile (a,σ,τ), its scheduling rates are determined in a single while loop (line 2-7), when the job profile is present in the interval being considered. Moreover, the scheduling rates are chosen to satisfy condition (87). Thus, given any $y \in [0,\tau]$ with $v(a,\sigma,\tau,y) > 0$, the expected service capacity at $a + \tau - y$ is given by

$$\mathbb{E}[P(a+\tau-y)] = \frac{I(t_1,t_2)}{t_2 - t_1}.$$

As this value does not depend on y, condition (i) holds.

Because the variable $\bar{v}(a,\sigma,\tau,y)$ is only updated in lines 1 and 6 in Algorithm 1, it can only take two values, *i.e.*

$$\bar{v}(a,\sigma,\tau,y) \in \{0,\infty\}. \tag{89}$$

Let $v(a,\sigma,\tau,y_1) > 0$, $v(a,\sigma,\tau,y_2) = 0$ for some y_1,y_2. Let $\bar{v}'(a,\sigma,\tau,y_1)$ be the value of $\bar{v}(a,\sigma,\tau,y_1)$ at the beginning of the loop in which $y_1 \in [t_1,t_2]$ in line 3. Similarly, let $\bar{v}'(a,\sigma,\tau,y_2)$ be the value of $\bar{v}(a,\sigma,\tau,y_2)$ at the beginning of the loop in which $y_2 \in [t_1,t_2]$ in line 3. From (88) and (89), there are two possible cases:

Case 1: $\bar{v}'(a,\sigma,\tau,y_1) = \bar{v}'(a,\sigma,\tau,y_2) = \infty$.

Case 2: $\bar{v}'(a,\sigma,\tau,y_1) = \infty, \bar{v}'(a,\sigma,\tau,y_2) = 0$.

In case 1, as the non-zero scheduling rates are determined in a single while loop (line 2-7), we have

$$\mathbb{E}[P(a+\tau-y_1)] = \mathbb{E}[P(a+\tau-y_2)] = \frac{I(t_1,t_2)}{t_2 - t_1},$$

which satisfies condition (i).

In case 2, it can be shown that

$$\mathbb{E}[P(a+\tau-y_1)] = \frac{I(t_1,t_2)}{t_2-t_1} \tag{90}$$

$$\leq \frac{I(t'_1,t'_2)}{t'_2-t'_1} \tag{91}$$

$$= \mathbb{E}[P(a+\tau-y_2)], \tag{92}$$

where $a+\tau-y_1 \in [t_1,t_2]$, and $a+\tau-y_2 \in [t'_1,t'_2]$. Here, (90) and (92) are due to (87), and (91) holds because time $a+\tau-y_2$ belongs to the interval that is dealt in an earlier while loop than time $a+\tau-y_1$. Combining (90)–(92) yields condition (ii).   Q.E.D.

Moreover, Algorithm 1 also minimizes the expected peaks in the service capacity.

Corollary 6. *Algorithm 1 is the solution for the following optimization problem:*

$$\underset{v:(1)(2)(3)(55)}{minimize} \ \underset{t\in\mathcal{T}}{\max} \mathbb{E}[P(t)].$$

Proof. Let $[t_1(1),t_2(1)]$ be the interval found by the first iteration in line 3 of Algorithm 1[17]. Note that $[t_1(1),t_2(1)]$ is the interval with largest job demand in an instance. Thus, the value for $\max_{t\in[t_1(1),t_2(1)]}\mathbb{E}[P(t)]$ is a lower bound for the value of $\max_{t\in\mathcal{T}}\mathbb{E}[P(t)]$. On the other hand, the minimum value for $\max_{t\in\mathcal{T}}\mathbb{E}[P(t)]$ equals $\frac{I(t_1(1),t_2(1))}{t_2(1)-t_1(1)}$. From line 4 in Algorithm 1, we have $\mathbb{E}[P(h)] = \frac{I(t_1(1),t_2(1))}{t_2(1)-t_1(1)}$ during $h \in [t_1(1),t_2(1))$, so Algorithm 1 forces $\max_{t\in[t_1(1),t_2(1)]}\mathbb{E}[P(t)]$ to take the minimum value.   Q.E.D.

5.4. Balancing stability and predictability

In the previous two sections, we show the optimal distributed policies that maximize each of predictability and stability separately. Beyond the two special cases, however, balancing stability and predictability is a much more complex problem, and it is too ambitious to seek a purely analytic solution. Instead, we characterize the Pareto-optimality condition for the distributed algorithm that balances predictability and stability in this section. Here, Pareto-optimality refers to the state when predictability cannot be improved without compromising stability or vice versa.

Recall that, with regard to maximizing predictability, it is favorable to have a fixed service rate over time (see Section 5.2). Meanwhile, with regards to maximizing stability, it is desirable to have a fixed service capacity over time (see Section 5.3). These special cases provide us with the intuition that the evenness of $r_k(t)$ and $\mathbb{E}[P(t)]$ may be used to balance predictability and stability. We formalize this intuition in the following theorem, which generalizes the result in Theorem 5 and Corollary 4.

Theorem 6. *The optimal solution of* (60), $v*$, *has the following two properties.*
(i) For each fixed job profile $(a;\sigma,\tau) \in \mathcal{T} \times S$, *if the optimal solution satisfies* $v^*(a,\sigma,\tau,a+\tau-t_1) > v^*(a,\sigma,\tau,a+\tau-t_2)$ *at some* $t_1 \neq t_2$, *then*

$$\alpha\mathbb{E}[P(t_1)] + \beta v^*(a,\sigma,\tau,a+\tau-t_1) \leq \alpha\mathbb{E}[P(t_2)] + \beta v^*(a,\sigma,\tau,a+\tau-t_2)$$

(ii) For each fixed job profile $(a;\sigma,\tau) \in \mathcal{T} \times S$, *if the optimal solution satisfies* $v^*(a,\sigma,\tau,a+\tau-t_1) = v^*(a,\sigma,\tau,a+\tau-t_2) \in (0,1)$ *at some* $t_1 \neq t_2$, *then*

$$\mathbb{E}[P(t_1)] = \mathbb{E}[P(t_2)]$$

---

[17] With a slight abuse of notation, we use (1) in $[t_1(1),t_2(1)]$ to denote the interval found in the first iteration

When $\alpha = 0$, Theorem 6 essentially states that Exact Scheduling maximizes predictability.This is because condition (ii) cannot happen when $\alpha = 0$, so the optimality condition reduces to the case when $v(a, \sigma, \tau, y)$ is constant at all $y \in [0, \tau]$. And the proof of Theorem 6 is an alternative way to derive Theorem 1. When $\beta = 0$, the conditions for $\alpha \mathbb{E}[P(h)] + \beta v(a, \sigma, \tau, a + \tau - h)$ reduces to the conditions stated in Corollary 4.

Proof of Theorem 6. Let $C_s(v), C_p(v)$ be defined as follows:

$$C_s(v) = \int_0^T \mathbb{E}[P(t)]^2 dt$$

$$C_p(v) = \int_0^T \text{Var}(P(t)) dt,$$

where the expected value and variance is taken under policy v. Note that the objective function of the optimization problem (60) equals

$$\lim_{T \to \infty} \frac{1}{T} \left\{ \alpha C_s(v) + \beta C_p(v) \right\}.$$

The constraints of (60) lead to

$$\int_{y=0}^{\tau} v(a, \sigma, \tau, y) dy = \sigma, \qquad\qquad (\sigma, \tau) \in S, \ a \in \mathcal{T} \qquad\qquad (93)$$

$$v(a, \sigma, \tau, y) \in [0, 1], \qquad\qquad (\sigma, \tau) \in S, \ a \in \mathcal{T}, \ y \in [0, \tau]. \qquad\qquad (94)$$

The Lagrangian associated with problem (60) is

$$\mathcal{L}(v; \mu, \nu) = \lim_{T \to \infty} \frac{1}{T} L(v; \mu, \nu)$$

where

$$L(v; \mu, \nu) = \alpha C_s(v) + \beta C_p(v) - \int_{a=0}^T \int_{(\sigma, \tau) \in S} \mu(a, \sigma, \tau) \int_{y=0}^{\tau} v(a, \sigma, \tau, y) dy d\sigma d\tau da$$

$$+ \int_{a=0}^T \int_{(\sigma, \tau) \in S} \int_{y=0}^{\tau} (\bar{\nu}(a, \sigma, \tau, y) - \underline{\nu}(a, \sigma, \tau, y)) v(a, \sigma, \tau, y) dy d\sigma d\tau da,$$

$\mu(a, \sigma, \tau)$ is the Lagrange multiplier associated with (93), and $\bar{\nu}(a, \sigma, \tau, y) \geq 0$ and $\underline{\nu}(a, \sigma, \tau, y) \geq 0$ are the Lagrange multipliers associated with the upper and lower bounds of $v(a, \sigma, \tau, y)$ in (94).

Let U be the space of scheduling policies of the form (55), and $U_f \subset U$ be the space of feasible scheduling policies, *i.e.*

$$U_f = \{v : v \text{ satisfies (93) and (94)}\}.$$

We can consider $L : U \to \mathbb{R}$ as a functional defined on U.

Now we impose an infinitesimal perturbation to v such that

$$v' = v + \epsilon \tilde{v} \in U_f.$$

In order to derive the optimality condition of (60), we will compute the values of

$$
\begin{aligned}
&L(v'; \mu, \nu) - L(v; \mu, \nu) \\
&= \alpha(C_s(v') - C_s(v)) + \beta(C_p(v') - C_p(v)) \\
&\quad + \int_{(\sigma, \tau) \in S} \int_{y=0}^{\tau} \int_{a=0}^{T} \epsilon(-\mu(a, \sigma, \tau) + \bar{\nu}(a, \sigma, \tau, y) - \underline{\nu}(a, \sigma, \tau, y)) \tilde{v}(a, \sigma, \tau, y) da dy d\sigma d\tau + O(\epsilon^2)
\end{aligned}
\qquad (95)
$$

First, we compute $C_s(v') - C_s(v)$. From Lemma 2, we have

$$C_s(v) = \int_{t=0}^{T} \left\{ \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} v(t+y-\tau,\sigma,\tau,y)\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau \right\}^2 dt, \tag{96}$$

which yields

$$C_s(v') = \int_{t=0}^{T} \left\{ \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} (v(t+y-\tau,\sigma,\tau,y) + \epsilon\tilde{v}(t+y-\tau,\sigma,\tau,y))\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau \right\}^2 dt$$

$$= 2\epsilon \int_{t=0}^{T} \left\{ \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} v(t+y-\tau,\sigma,\tau,y)\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau \right. \tag{97}$$

$$\int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} \tilde{v}(t+y-\tau,\sigma,\tau,y)\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau \Big\} dt$$

$$+ \int_{t=0}^{T} \left\{ \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} v(t+y-\tau,\sigma,\tau,y)\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau \right\}^2 dt + O(\epsilon^2)$$

$$= 2\epsilon \int_{t=0}^{T} \left\{ \int_{(\sigma,\tau)\in S} \int_{z=t-\tau}^{t} v(z,\sigma,\tau,z-t+\tau)\Lambda(z,\sigma,\tau)dzd\sigma d\tau \right. \tag{98}$$

$$\int_{(\sigma,\tau)\in S} \int_{z=t-\tau}^{t} \tilde{v}(z,\sigma,\tau,z-t+\tau)\Lambda(z,\sigma,\tau)dzd\sigma d\tau \Big\} dt$$

$$+ \int_{t=0}^{T} \left\{ \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} v(t+y-\tau,\sigma,\tau,y)\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau \right\}^2 dt + O(\epsilon^2)$$

$$= 2\epsilon \int_{t=0}^{T} G(t;v) \int_{(\sigma,\tau)\in S} \int_{z=t-\tau}^{t} \tilde{v}(z,\sigma,\tau,z-t+\tau)\Lambda(z,\sigma,\tau)dzd\sigma d\tau dt \tag{99}$$

$$+ \int_{t=0}^{T} \left\{ \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} v(t+y-\tau,\sigma,\tau,y)\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau \right\}^2 dt + O(\epsilon^2).$$

Here, (98) is due to the change of variable in integration from (σ,τ,y) to $(z=t+y-\tau,\sigma,\tau)$. In (99), the function $G(t;v)$ is defined to be

$$G(t;v) = \int_{(\sigma,\tau)\in S} \int_{a=t-\tau}^{t} v(\sigma,\tau,a,y)\Lambda(a,\sigma,\tau)dad\sigma d\tau. \tag{100}$$

Combining (96) and (97)–(99), we have

$$C_s(v') - C_s(v) = 2\epsilon \int_{t=0}^{T} \int_{(\sigma,\tau)\in S} \int_{z=t-\tau}^{t} G(t;v)\tilde{v}(z,\sigma,\tau,z-t+\tau)\Lambda(z,\sigma,\tau)dzd\sigma d\tau dt + O(\epsilon^2)$$

$$= 2\epsilon \int_{t=0}^{T} \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} G(t;v)\tilde{v}(t-\tau+y,\sigma,\tau,y)\Lambda(z,\sigma,\tau)dyd\sigma d\tau dt + O(\epsilon^2)$$

$$= 2\epsilon \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} \int_{t=0}^{T} G(t;v)\tilde{v}(t-\tau+y,\sigma,\tau,y)\Lambda(z,\sigma,\tau)dtdyd\sigma d\tau + O(\epsilon^2)$$

$$= 2\epsilon \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} \int_{a=0}^{T} G(a+\tau-y;v)\tilde{v}(a,\sigma,\tau,y)\Lambda(a,\sigma,\tau)dadyd\sigma d\tau + O(\epsilon^2) \tag{101}$$

Second, $C_p(v') - C_p(v)$ can be computed as

$$C_p(v) = \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} \int_{t=0}^{T} v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dtdyd\sigma d\tau$$

$$= \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} \int_{a=y-\tau}^{T+y-\tau} v(a,\sigma,\tau,y)^2 \Lambda(a,\sigma,\tau) da\,dy\,d\sigma\,d\tau \tag{102}$$

$$= \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} \int_{a=0}^{T} v(a,\sigma,\tau,y)^2 \Lambda(a,\sigma,\tau) da\,dy\,d\sigma\,d\tau \tag{103}$$

where (102) is due to the change of variable in integration from (σ,τ,y,t) to $(\sigma,\tau,y,a=t+y-\tau)$, and (103) is due to assumptions (51) and (52). From (103), we have

$$C_p(v') - C_p(v) = 2\epsilon \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} \int_{a=0}^{T} v(a,\sigma,\tau,y)\tilde{v}(a,\sigma,\tau,y)\Lambda(a,\sigma,\tau) da\,dy\,d\sigma\,d\tau + O(\epsilon^2). \tag{104}$$

Combining (95), (101), and (104) yields

$$L(v';\mu,\nu) - L(v;\mu,\nu) = \epsilon \int_{(\sigma,\tau)\in S} \int_{y=0}^{\tau} \int_{a=0}^{T} F_v(a,\sigma,\tau,y)\tilde{v}(a,\sigma,\tau,y) da\,dy\,d\sigma\,d\tau + O(\epsilon^2),$$

where

$$F_v(a,\sigma,\tau,y) = 2(\alpha G(a+\tau-y;v) + \beta v(a,\sigma,\tau,y))\Lambda(a,\sigma,\tau) - \mu(a,\sigma,\tau) + \bar{\nu}(a,\sigma,\tau,y) - \underline{\nu}(a,\sigma,\tau,y))$$

$$= 2(\alpha \mathbb{E}_v[P(a+\tau-y)] + \beta v(a,\sigma,\tau,y))\Lambda(a,\sigma,\tau) - \mu(a,\sigma,\tau) + \bar{\nu}(a,\sigma,\tau,y) - \underline{\nu}(a,\sigma,\tau,y)).$$

Here, the second equality is due to (100) and the observation that[18]

$$\int_{(\sigma,\tau)\in S} \int_{a=t-\tau}^{t} v(\sigma,\tau,a,y)\Lambda(a,\sigma,\tau) da\,d\sigma\,d\tau = \mathbb{E}_v[P(t)].$$

For the functional L to be stationary at $v^* \in U_f$, the first term should be zero for any $\tilde{v}(a,\sigma,\tau,y)$ satisfying the constraint $v' \in U_f$. This condition is given by

$$\alpha\mathbb{E}[P(t)] + \beta v^*(a,\sigma,\tau,a+\tau-y) = \frac{\mu(a,\sigma,\tau) - \bar{\nu}(a,\sigma,\tau,y) + \underline{\nu}(a,\sigma,\tau,y)}{2\Lambda(a,\sigma,\tau)} \tag{105}$$

for any $y \in [0,\tau]$, $(\sigma,\tau) \in S$, $a \in \mathcal{T}$. Since the optimal solution of (59) is a stationary point of L, (105) is the necessary condition for optimality.

Now we are ready to derive properties (i) and (ii). When $v^*(a,\sigma,\tau,y_1) > v^*(a,\sigma,\tau,y_2)$ at some $y_1 \neq y_2$, there are four possible cases:

Case 1: $v^*(a,\sigma,\tau,y_1), v^*(a,\sigma,\tau,y_2) \in (0,1)$

Case 2: $v^*(a,\sigma,\tau,y_1) \in (0,1)$ and $v^*(a,\sigma,\tau,y_2) = 0$

Case 3: $v^*(a,\sigma,\tau,y_1) = 1$ and $v^*(a,\sigma,\tau,y_2) = 0$

Case 4: $v^*(a,\sigma,\tau,y_1) = 1$ and $v^*(a,\sigma,\tau,y_2) \in (0,1)$

For case 1, we have

$$\alpha\mathbb{E}[P(a+\tau-y_1)] + \beta v^*(a,\sigma,\tau,y_1) = \frac{\mu(a,\sigma,\tau)}{2\Lambda(a,\sigma,\tau)}$$
$$= \alpha\mathbb{E}[P(a+\tau-y_2)] + \beta v^*(a,\sigma,\tau,y_2).$$

---

[18] The subscript v in $\mathbb{E}$ denotes the expected value under policy v.

For case 2, we have

$$
\begin{aligned}
\alpha\mathbb{E}[P(a+\tau-y_1)]+\beta v^*(a,\sigma,\tau,y_1) &= \frac{\mu(a,\sigma,\tau)}{2\Lambda(a,\sigma,\tau)} \\
&\leq \frac{\mu(a,\sigma,\tau)+\underline{\nu}(a,\sigma,\tau,y_2)}{2\Lambda(a,\sigma,\tau)} \\
&= \alpha\mathbb{E}[P(a+\tau-y_2)]+\beta v(a,\sigma,\tau,y_2).
\end{aligned}
$$

For case 3, we have

$$
\begin{aligned}
\alpha\mathbb{E}[P(a+\tau-y_1)]+\beta v(a,\sigma,\tau,y_1) &= \frac{\mu(a,\sigma,\tau)-\bar{\nu}(a,\sigma,\tau,y_1)}{2\Lambda(a,\sigma,\tau)} \\
&\leq \frac{\mu(a,\sigma,\tau)+\underline{\nu}(a,\sigma,\tau,y_2)}{2\Lambda(a,\sigma,\tau)} \\
&= \alpha\mathbb{E}[P(a+\tau-y_2)]+\beta v(a,\sigma,\tau,y_2).
\end{aligned}
$$

For case 4, we have

$$
\begin{aligned}
\alpha\mathbb{E}[P(a+\tau-y_1)]+\beta v(a,\sigma,\tau,y_1) &= \frac{\mu(a,\sigma,\tau)-\bar{\nu}(a,\sigma,\tau,y_1)}{2\Lambda(a,\sigma,\tau)} \\
&\leq \frac{\mu(a,\sigma,\tau)}{2\Lambda(a,\sigma,\tau)} \\
&= \alpha\mathbb{E}[P(a+\tau-y_2)]+\beta v(a,\sigma,\tau,y_2).
\end{aligned}
$$

The above conditions for cases 1-4 yield property (i).

If $v^*(a,\sigma,\tau,y_1)=v^*(a,\sigma,\tau,y_2)\in(0,1)$, then

$$
\begin{aligned}
\alpha\mathbb{E}[P(a+\tau-y_1)]+\beta v^*(a,\sigma,\tau,y_1) &= \frac{\mu(a,\sigma,\tau)}{2\Lambda(a,\sigma,\tau)} \\
&= \alpha\mathbb{E}[P(a+\tau-y_2)]+\beta v^*(a,\sigma,\tau,y_2),
\end{aligned}
$$

implying $\mathbb{E}[P(a+\tau-y_1)]=\mathbb{E}[P(a+\tau-y_2)]$. Thus, property (ii) holds.   Q.E.D.

6. Conclusion

As it becomes more common for service systems to have a dynamic capacity that instantaneously adapts to demand, the goal of providing a high quality of service (*e.g.* meeting deadlines) while minimizing the variance of service capacity has received increasing attention. Though there exists extensive literature analyzing existing algorithms, few analytic results characterizing optimal policies were known in such settings.

In this paper, we characterize the optimal distributed policies in many common scenarios, stationary and non-stationary arrivals, strict and soft demands, with and without deadline extensions, and a variety of objective functions. The results highlight that novel generalizations of Exact Scheduling maximize the predictability (*i.e.* minimizes the service capacity variance) under both stationary and non-stationary Poisson arrival processes. The derivation of the results is enabled by a combination of a few ideas. To obtain optimal algorithms, we need to solve functional optimization problems whose constraints include a PDE (the continuity equation). Instead of directly solving these problems, we consider their relaxations without the PDE constraints, which extend the class of admissible scheduling policies. We first solve the relaxed problem to provide lower bounds to their minimum costs. The constraint sets of the original optimization problems are subsets of the relaxed problems, so the optimal solutions of the relaxed problems do not necessarily lie within the original constraint sets. However, interestingly, for a few different settings considered

in this paper, the optimal schedulers from the latter problem happen to lie within the constraint set of the original optimization problems, thereby also optimal for the original problems.

When the goal is to balance stability and predictability, more complex policies turn out to be optimal. Such optimal distributed policies include a stochastic counterpart of the YDS algorithm, which is shown to be the optimal algorithm in a deterministic setting. This similarity in algorithm structures and the proofs suggest a new bridge between the stochastic and worst-case scheduling communities. This connection will be interesting to explore in future work.

In addition to characterizing optimal distributed policies, we also bound the gap between the performance of distributed policies and centralized policies using both theory and experiment. To derive theoretical bounds, we adopt optimal control techniques, which can also be extended to derive performance bounds for related problems. In an experiment conducted in the electrical vehicle charging testbed, we observe that the optimal distributed policies could nearly match the performance of centralized policies.

Going forward, we note two interesting future directions of this paper:

Non-asymptotic optimization in related problems. Typically, the analysis of scheduling policies for non-stationary settings with deadlines has been done using asymptotic regimes, *e.g.* heavy-traffic regimes. However, the techniques we develop in this paper do not require asymptotic approximations. Thus, in addition to the results we have proven, our techniques are also an important contribution. We hope these techniques will inspire the discovery of other optimality results in applications that involve scheduling.

The design space of partially-centralized schedulers. In Section 4.1, we made an initial attempt to explore the middle ground between fully-centralized algorithms and fully-distributed algorithms. We demonstrated numerically that Exact Scheduling PC can have a competitive performance to fully-centralized algorithms while preserving the scalability of Exact Scheduling, which is fully-distributed. Beyond our initial analysis, the vast design space of partially-centralized algorithms may have great potential in balancing performance versus scalability.

Acknowledgments

References

[1] John A Stankovic and Krithi Ramamritham. What is predictability for real-time systems? *Real-Time Systems*, 2(4):247–254, 1990.

[2] Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar, Matt Tolton, and Theo Vassilakis. Dremel: interactive analysis of web-scale datasets. *Proceedings of the VLDB Endowment*, 3(1-2):330–339, 2010.

[3] Giorgio C Buttazzo. *Hard real-time computing systems: predictable scheduling algorithms and applications*, volume 24. Springer Science & Business Media, 2011.

[4] Kathleen Spees and Lester B Lave. Demand response and electricity market efficiency. *The Electricity Journal*, 20(3):69–85, 2007.

[5] Mahdi Behrangrad. A review of demand side management business models in the electricity market. *Renewable and Sustainable Energy Reviews*, 47:270–283, 2015.

[6] Nikhil Bansal, Tracy Kimbrel, and Kirk Pruhs. Speed scaling to manage energy and temperature. *Journal of the ACM (JACM)*, 54(1):3, 2007.

[7] Shivendra S Panwar, Don Towsley, and Jack K Wolf. Optimal scheduling policies for a class of queues with customer deadlines to the beginning of service. *Journal of the ACM (JACM)*, 35(4):832–844, 1988.

[8] Shivendra S Panwar and Don Towsley. On the optimality of the ste rule for multiple server queues that serve customers with deadlines. Technical report, University of Massachusetts, 1988.

[9] Partha P Bhattacharya and Anthony Ephremides. Optimal scheduling with strict deadlines. *IEEE Transactions on Automatic Control*, 34(7):721–728, 1989.

[10] John P Lehoczky. Using real-time queueing theory to control lateness in real-time systems. *ACM SIGMETRICS Performance Evaluation Review*, 25(1):158–168, 1997.

[11] H Christian Gromoll and Łukasz Kruk. Heavy traffic limit for a processor sharing queue with soft deadlines. *The Annals of Applied Probability*, 17(3):1049–1101, 2007.

[12] Andres Ferragut, Fernando Paganini, and Adam Wierman. Controlling the variability of capacity allocations using service deferrals. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 2(3):15, 2017.

[13] Chung Laung Liu and James W Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, 20(1):46–61, 1973.

[14] Frances Yao, Alan Demers, and Scott Shenker. A scheduling model for reduced cpu energy. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 374–382. IEEE, 1995.

[15] George Lee, Ted Lee, Zhi Low, Steven H Low, and Christine Ortega. Adaptive charging network for electric vehicles. In *Signal and Information Processing (GlobalSIP), 2016 IEEE Global Conference on*, pages 891–895. IEEE, 2016.

[16] John A Stankovic, Marco Spuri, Krithi Ramamritham, and Giorgio C Buttazzo. *Deadline scheduling for real-time systems: EDF and related algorithms*, volume 460. Springer Science & Business Media, 2012.

[17] Jiawei Hong, Xiaonan Tan, and Don Towsley. A performance analysis of minimum laxity and earliest deadline scheduling in a real-time system. *IEEE Transactions on Computers*, 38(12):1736–1744, 1989.

[18] Michael Pinedo. Stochastic scheduling with release dates and due dates. *Operations Research*, 31(3):559–572, 1983.

[19] John P Lehoczky. Real-time queueing network theory. In *Proc of the 18th IEEE Real-Time Systems Symposium*, pages 58–67, 1997.

[20] Martin Zeballos, Andres Ferragut, and Fernando Paganini. Proportional fairness for ev charging in overload. *IEEE Transactions on Smart Grid*, 10(6):6792–6801, 2019.

[21] Richard L Daniels, Barbara J Hoopes, and Joseph B Mazzola. Scheduling parallel manufacturing cells with resource flexibility. *Management science*, 42(9):1260–1276, 1996.

[22] C-Y Lee and V Jorge Leon. Machine scheduling with a rate-modifying activity. *European Journal of Operational Research*, 128(1):119–128, 2001.

[23] Jeffrey Dean and Luiz André Barroso. The tail at scale. *Communications of the ACM*, 56(2):74–80, 2013.

[24] Eric Boutin, Jaliya Ekanayake, Wei Lin, Bing Shi, Jingren Zhou, Zhengping Qian, Ming Wu, and Lidong Zhou. Apollo: Scalable and coordinated scheduling for cloud-scale computing. In *OSDI*, volume 14, pages 285–300, 2014.

[25] Xiaorui Wang and Ming Chen. Cluster-level feedback power control for performance optimization. In *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*, pages 101–110. IEEE, 2008.

[26] Minghong Lin, Adam Wierman, Lachlan LH Andrew, and Eno Thereska. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Transactions on Networking*, 21(5):1378–1391, 2013.

[27] Dara Kusic, Jeffrey O Kephart, James E Hanson, Nagarajan Kandasamy, and Guofei Jiang. Power and performance management of virtualized computing environments via lookahead control. *Cluster computing*, 12(1):1–15, 2009.

[28] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H Low, and Lachlan LH Andrew. Greening geographical load balancing. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 233–244. ACM, 2011.

[29] Anshul Gandhi. *Dynamic server provisioning for data center power management*. PhD thesis, Carnegie Mellon University, 2013.

[30] Anshul Gandhi, Mor Harchol-Balter, Rajarshi Das, and Charles Lefurgy. Optimal power allocation in server farms. In *ACM SIGMETRICS Performance Evaluation Review*, volume 37(1), pages 157–168. ACM, 2009.

[31] Ashutosh Nayyar, Josh Taylor, Anand Subramanian, Kameshwar Poolla, and Pravin Varaiya. Aggregate flexibility of a collection of loadsπ. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 5600–5607. IEEE, 2013.

[32] Anand Subramanian, Manuel J Garcia, Duncan S Callaway, Kameshwar Poolla, and Pravin Varaiya. Real-time scheduling of distributed resources. *IEEE Transactions on Smart Grid*, 4(4):2122–2130, 2013.

[33] Lingwen Gan, Ufuk Topcu, and Steven H Low. Optimal decentralized protocol for electric vehicle charging. *IEEE Transactions on Power Systems*, 28(2):940–951, 2013.

[34] Niangjun Chen, Lingwen Gan, Steven H Low, and Adam Wierman. Distributional analysis for model predictive deferrable load control. In *Proc. of the IEEE 53rd annual Conference on Decision and Control*, 2014.

[35] François Baccelli and Bartlomiej Błaszczyszyn. *Stochastic Geometry and Wireless Networks, Volume I - Theory*. Now Publishers, 2009.

[36] Albert Greenberg, James Hamilton, David A Maltz, and Parveen Patel. The cost of a cloud: research problems in data center networks. *ACM SIGCOMM computer communication review*, 39(1):68–73, 2008.

[37] Peter Brucker and P Brucker. *Scheduling algorithms*, volume 3. Springer, 2007.

[38] Brendan Lucier, Ishai Menache, Joseph Naor, and Jonathan Yaniv. Efficient online scheduling for deadline-sensitive jobs. In *Proceedings of the twenty-fifth annual ACM symposium on Parallelism in algorithms and architectures*, pages 305–314, 2013.

[39] Joseph Pedlosky. *Geophysical fluid dynamics*. Springer Science & Business Media, 2013.

[40] Francois Baccelli and Pierre Brémaud. *Elements of queueing theory: Palm Martingale calculus and stochastic recurrences*, volume 26. Springer Science & Business Media, 2013.

[41] John Lehoczky, Lui Sha, and Ye Ding. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. In *Real Time Systems Symposium, 1989., Proceedings.*, pages 166–171. IEEE, 1989.

[42] Shelby L Brumelle. On the relation between customer and time averages in queues. *Journal of Applied Probability*, 8(3):508–520, 1971.

[43] Jianzhong Du and Joseph Y-T Leung. Minimizing total tardiness on one machine is np-hard. *Mathematics of operations research*, 15(3):483–495, 1990.

[44] Kenneth R Baker and Gary D Scudder. Sequencing with earliness and tardiness penalties: a review. *Operations research*, 38(1):22–36, 1990.

[45] Wanrong Tang and Ying Jun Zhang. A model predictive control approach for low-complexity electric vehicle charging scheduling: Optimality and scalability. *IEEE transactions on power systems*, 32(2):1050–1063, 2016.

[46] HS Das, MM Rahman, S Li, and CW Tan. Electric vehicles standards, charging infrastructure, and impact on grid integration: A technological review. *Renewable and Sustainable Energy Reviews*, 120:109618, 2020.

[47] Lingwen Gan, Adam Wierman, Ufuk Topcu, Niangjun Chen, and Steven H Low. Real-time deferrable load control: handling the uncertainties of renewable generation. In *Proceedings of eEnergy*, 2013.

[48] Thomas M Cover and Joy A Thomas. *Elements of information theory*. Wiley-Interscience, 2nd edition edition, July 18 2006.

[49] J. Yang and S. Ulukus. Optimal packet scheduling in a multiple access channel with energy harvesting transmitters. *Journal of Communications and Networks*, 14(2):140–150, April 2012.

[50] Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 2. Athena scientific Belmont, MA, 1995.

Appendix A: Proof of Lemma 2

In this section, we present results that are useful for proving our main theorems. First, we restate one part of the Campbell's theorem, which is relevant to our proofs.

Theorem 7 (Campbell formula for marked Poisson processes [40]). *Consider an independently marked Poisson point process $\{x_k\} \subset \mathbb{R}^d$ with intensity measure $\Lambda : \mathbb{R}^d \to \mathbb{R}_+$ and marks in $\mathbb{R}^p$ with distribution $F(dz)$. Let $g : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$ be a measurable function satisfying*

$$\int_{\mathbb{R}^d} \int_{\mathbb{R}^p} g(x,z)^2 \Lambda(dx) F(dz) < \infty.$$

Then, the random sum

$$G = \sum_{k \in \mathbb{Z}} g(x_k, z_k)$$

is absolutely convergent with probability one and satisfies

$$\mathbb{E}[G] = \int_{\mathbb{R}^d} \int_{\mathbb{R}^p} g(x,z) \Lambda(dx) F(dz)$$

$$\mathrm{Var}(G) = \int_{\mathbb{R}^d} \int_{\mathbb{R}^p} g(x,z)^2 \Lambda(dx) F(dz).$$

Throughout, we consider a scheduling policy (14), which is defined by a function $v : S \times \mathbb{R} \to \mathbb{R}_+$ as follows:

$$r_k(t) = v(\sigma_k, \tau_k, y_k(t)) \qquad\qquad\qquad k \in \mathcal{V}.$$

The function v satisfies

$$\int_{a_k}^{\infty} v(\sigma_k, \tau_k, a_k + \tau_k - t)^2 dt = \int_{a_k}^{\infty} r_k(t)^2 dt \le \sigma_k$$

where the maximum value of the integral is attained by Immediate Scheduling $r_k(t) = 1\{t \in [a_k, a_k + \sigma_k]\}$ subject to constraint (1). Therefore, we have

$$\int_{(\sigma,\tau) \in S} \int_{\mathbb{R}} v(\sigma, \tau, y)^2 dy \Lambda f(\sigma, \tau) d\sigma d\tau \le \Lambda \int_{(\sigma,\tau) \in S} \sigma f(\sigma, \tau) d\sigma d\tau = \mathbb{E}[\sigma] \Lambda < \infty \qquad (106)$$

Combining (106) and Theorem 7, we obtain (62) and (63):

$$\mathbb{E}[P(t)] = \int_{(\sigma,\tau) \in S} \int_0^{\tau} v(\sigma, \tau, y) \Lambda f(\sigma, \tau) dy d\sigma d\tau$$

$$\mathrm{Var}(P(t)) = \int_{(\sigma,\tau) \in S} \int_0^{\tau} v(\sigma, \tau, y)^2 \Lambda f(\sigma, \tau) dy d\sigma d\tau,$$

which yields Lemma 2.

Appendix B: Proof of Proposition 1

We observe that

$$\int_{-\infty}^{\infty} \int_0^{\infty} \frac{\partial}{\partial y} \lambda(x,y) x \, dx \, dy = \int_0^{\infty} x \left\{ \int_{-\infty}^{\infty} \frac{\partial}{\partial y} \lambda(x,y) dy \right\} dx \qquad (107)$$

$$= -\int_0^{\infty} x \lim_{L \to \infty} \lambda(x,L) dx$$

$$= -\Lambda \mathbb{E}[\sigma - \hat{\sigma}(\sigma, \tau)] \qquad (108)$$

where (107) holds because bounded S implies that $\lambda(x, \infty) = 0$. Therefore, the stationary mean of the service capacity satisfies

$$\mathbb{E}[P(t)] = \mathbb{E}\left[\sum_{k \in \mathcal{V}} u(x_k(t), y_k(t)) \right]$$

$$= \int_{-\infty}^{\infty} \int_{0}^{\infty} \lambda(x,y)u(x,y)dxdy$$

$$= -\int_{-\infty}^{\infty} \int_{0}^{\infty} \frac{d}{dx}\left(\lambda(x,y)u(x,y)\right)xdxdy \tag{109}$$

$$= \int_{-\infty}^{\infty} \int_{0}^{\infty} \left(\frac{\partial}{\partial y}\lambda(x,y) + \Lambda f(x,y)\right)xdxdy \tag{110}$$

$$= -\Lambda\mathbb{E}[\sigma - \hat{\sigma}(\sigma,\tau)] + \Lambda\mathbb{E}[\sigma] \tag{111}$$

$$= \Lambda\mathbb{E}[\hat{\sigma}(\sigma,\tau)]$$

Here, (109) is due to Integration by Parts, (110) is due to (5), (111) is due to (107)–(108).

Appendix C: Proof of Theorem 2

Since the constraints of (7) is hard to solve, we first consider providing a lower bound on its optimal solution. Again, we consider the class of control policies representable by (14) and the optimization problem

$$\underset{v:(1)(3)(14)}{\text{minimize}} \ \text{Var}(P) + \mathbb{E}[U]. \tag{112}$$

Because the constraint set of (112) contains that of (7), the optimal value of (112) lower-bounds the optimal value of (7). Therefore, to prove Theorem 2, it suffices to solve (112) (in the next lemma) and observe that its optimal solution is representable by a control policy of the form (4).

Lemma 7. *The optimal solution of* (112) *is*

$$v(\sigma,\tau,y) = \min\left\{\frac{\delta}{2}, \frac{\sigma}{\tau}\right\}\mathbb{1}\left\{y > 0\right\}, \tag{113}$$

and it achieves the optimal value (22).

Proof.   First, we derive an analytical formula for $\mathbb{E}[U]$ as a function of the scheduling policy v. Let

$$\hat{\sigma}(\sigma,\tau) = \int_{-\infty}^{\tau} v(\sigma,\tau,y)dy, \tag{114}$$

be the actual amount of service received by a job with demand σ and sojourn time τ. The amount of unsatisfied demand for this job is $\sigma - \hat{\sigma}(\sigma,\tau)$. Additionally, $\hat{\sigma}(\sigma,\tau)$ satisfies

$$0 \leq \hat{\sigma}(\sigma,\tau) \leq \sigma, \qquad\qquad \forall(\sigma,\tau) \in S. \tag{115}$$

Consequently, the stationary mean of U satisfies

$$\mathbb{E}[U] = \lim_{t\to\infty} \mathbb{E}\left[\sum_{k\in\mathcal{V}:a_k+\tau_k=t} (\sigma_k - \hat{\sigma}(\sigma_k,\tau_k))\right]$$

$$= \int_{(\sigma,\tau)\in S} (\sigma - \hat{\sigma}(\sigma,\tau))\Lambda f(\sigma,\tau)d\sigma d\tau \tag{116}$$

Then, we use (116) to rewrite (112) as follows

$$\inf_{v:(1)(3)(14)} \ \text{Var}(P) + \mathbb{E}[U]$$

$$= \inf_{\hat{\sigma}:(115)}\left[\inf_{v:(1)(3)(14)(114)} \text{Var}(P) + \delta\int_{(\sigma,\tau)\in S}(\sigma - \hat{\sigma}(\sigma,\tau))\Lambda f(\sigma,\tau)d\sigma d\tau\right] \tag{117}$$

$$= \inf_{\hat{\sigma}:(115)}\left[\left\{\inf_{v:(1)(3)(14)(114)} \text{Var}(P)\right\} + \delta\int_{(\sigma,\tau)\in S}(\sigma - \hat{\sigma}(\sigma,\tau))\Lambda f(\sigma,\tau)d\sigma d\tau\right]. \tag{118}$$

Equality (118) holds because, constrained on $\hat{\sigma}(\sigma,\tau) = \int_0^\tau v(\sigma,y,\tau)dy$ for some fixed $\hat{\sigma}$, the second term of (117) is not a function of v. From Lemma 1, the first term of (118) admits the closed-form expression

$$\inf_{v:(1)(3)(14)} \text{Var}(P) = \int_{(\sigma,s)\in S} \frac{\hat{\sigma}(\sigma,\tau)^2}{\tau}\Lambda f(\sigma,\tau)d\sigma d\tau, \tag{119}$$

which is attained by

$$v(\sigma, \tau, y) = \frac{\hat{\sigma}(\sigma, \tau)}{\tau}. \tag{120}$$

Substitute (119) into (118) yields

$$\inf_{\hat{\sigma}:(115)} \int_{(\sigma,\tau) \in S} \left\{ \frac{\hat{\sigma}(\sigma,\tau)^2}{\tau} + \delta(\sigma - \hat{\sigma}(\sigma,\tau)) \right\} \Lambda f(\sigma,\tau) d\sigma d\tau, \tag{121}$$

where the optimization variable is now $\hat{\sigma}$ instead of v. To derive a closed-form solution of (112), we can minimize the integrand of (121) point-wisely. By doing so, we observe that, for each $(\sigma, \tau) \in S$, a necessary and sufficient condition for optimality is

$$\hat{\sigma}(\sigma, \tau) = \arg \inf_{\hat{\sigma}:(115)} \frac{\hat{\sigma}(\sigma,\tau)^2}{\tau} + \delta(\sigma - \hat{\sigma}(\sigma,\tau)) = \min \left\{ \frac{\delta\tau}{2}, \sigma \right\}. \tag{122}$$

Combining (120) and (122), we obtain that (113) is the optimal solution of (112). Substitute (113) into (121), we obtain its optimal value (22).   Q.E.D.

Given Lemma 7, Theorem 2 can be derived as follows. It can be verified that scheduler (113) can be realized as (21) using a scheduling policy of the form (21). This implies that the optimal solution of problem (112) also lies within the constraint set of problem (7). Because the cost attained by scheduler (113) is a lower bound on the optimal value of problem (7), the optimal solution of problem (7) is scheduler (21).

Appendix D: Proof of Theorem 3

Since the constraints of (8) is hard to solve, we first consider providing a lower bound on its optimal solution. Again, we consider the class of control policies representable by (14) and the optimization problem

$$\underset{v:(1)(2)(14)}{\text{minimize}} \quad \text{Var}(P) + \mathbb{E}[W]. \tag{123}$$

Because the optimal value of (123) lower-bounds that of (7), to prove Theorem 3, we can solve (123) (in the next lemma) and observe that its optimal solution is representable by a control policy of the form (4).

Lemma 8. *The optimal solution of* (123) *is*

$$v(\sigma, \tau, y) = \begin{cases} \dfrac{\sigma}{\tau} \mathbb{1}\{y > 0\} & \text{if } \dfrac{\sigma}{\tau} \leq \sqrt{\epsilon} \\ \sqrt{\epsilon} \, \mathbb{1}\left\{ y > \tau - \dfrac{\sigma}{\sqrt{\epsilon}} \right\} & \text{otherwise} \end{cases}. \tag{124}$$

and it achieves the optimal value (24).

Proof. With a slight abuse of notation, let

$$\hat{\tau}(\sigma, \tau) = \begin{cases} \tau & \text{if } v(\sigma, \tau, y) = 0, \, \forall y < 0 \\ \tau - \min\{\bar{y} : v(\sigma, \tau, y) = 0, \forall y \leq \bar{y}\} & \text{otherwise} \end{cases} \tag{125}$$

denote the actual sojourn time for jobs having a service demand σ and a sojourn time τ. Then, the stationary mean of W satisfies

$$\mathbb{E}[W] = \epsilon \int_{(\sigma,\tau) \in S} (\hat{\tau}(\sigma,\tau) - \tau) \Lambda f(\sigma,\tau) d\sigma d\tau.$$

The optimization problem (123) can then be written into

$$\inf_{v:(1)(2)(14)} \text{Var}(P) + \mathbb{E}[\epsilon W]$$

$$= \inf_{\hat{\tau} \geq \tau} \left[\left\{ \inf_{v:(1)(2)(14)} \text{Var}(P) \right\} + \epsilon \int_{(\sigma,\tau) \in S} (\hat{\tau}(\sigma,\tau) - \tau) \Lambda f(\sigma,\tau) d\sigma d\tau \right] \tag{126}$$

$$= \inf_{\hat{\tau} \geq \tau} \int_{(\sigma,\tau) \in S} \left\{ \frac{\sigma^2}{\hat{\tau}} + \epsilon(\hat{\tau}(\sigma,\tau) - \tau) \right\} \Lambda f(\sigma,\tau) d\sigma d\tau, \tag{127}$$

where $\inf_{v:(1)(2)(14)} \text{Var}(P)$ in (126) is attained by

$$v(\sigma, \tau, y) = \frac{\sigma}{\hat{\tau}(\sigma, \tau)}. \tag{128}$$

The optimal choice of deadline extensions $\hat{\tau}^\star(\sigma, \tau)$ is the point-wise maximum of the integrand of (127), *i.e.*

$$\hat{\tau}^\star(\sigma, \tau) = \arg\inf_{\hat{\sigma}:(115)} \frac{\sigma^2}{\hat{\tau}} + \epsilon(\hat{\tau}(\sigma, \tau) - \tau) = \left\{ \frac{\sigma}{\sqrt{\epsilon}}, \tau \right\}. \tag{129}$$

Combining (128) and (129), we obtain (124) as the closed-form solution of (123). Q.E.D.

Given Lemma 8, we are now ready to prove Theorem 3.

Proof (Theorem 3) Recall that the optimal value of problem (123) lower-bounds the optimal value of problem (8). Therefore, if there is a policy of the form (4) that produces identical service rates to (124), it is also optimal for problem (8). Next, we show that the policy (23) satisfies the above description.

Given any job $k \in \mathcal{V}$ with $\sigma \leq \tau\sqrt{\epsilon}$, both (23) and (124) produce the service rates $r_k(t) = \sigma_k/\tau_k$ if $t \in [a_k, a_k + \tau_k]$ and $r_k(t) = 0$ otherwise. Given any job $k \in \mathcal{V}$ with $\sigma > \sqrt{\epsilon}\tau$, (124) produces the service rates $r_k(t) = \sqrt{\epsilon}$ if $t \in [a_k, a_k + \sigma/\sqrt{\epsilon}]$ and $r_k(t) = 0$ otherwise. Observe that under the policy (23), for any $y(t) > 0$, we have

$$\frac{x(t)}{y(t)} - \frac{\sigma}{\tau} = \frac{\sigma - \sqrt{\epsilon}(t-a)}{\tau - (t-a)} - \frac{\sigma}{\tau} \geq \frac{(\sigma/\tau - \sqrt{\epsilon})(t-a)}{\tau - (t-a)} \geq 0,$$

where the third inequality is due to $\tau \geq \sqrt{\epsilon}$. Thus, the policy (124) also produce the service rates $r_k(t) = \sqrt{\epsilon}$ if $t \in [a_k, a_k + \sigma/\sqrt{\epsilon}]$ and $r_k(t) = 0$ otherwise. Q.E.D.

Appendix E: Proof of Theorem 4

We first consider providing a lower bound of problem (9) by solving the optimization problem

$$\min_{v:(1)(14)} \text{Var}(P) + \mathbb{E}[U] + \mathbb{E}[W]. \tag{130}$$

The solution of problem (130) is given in the next lemma, which is also a feasible policy for the constraint set of problem (9).

Lemma 9. *The optimal solution of problem* (130) *is*

$$v(\sigma, \tau, y) = \begin{cases} \frac{\sigma}{\tau} 1\{y > 0\} & \text{if } \frac{\sigma}{\tau} \leq \min\left\{\frac{\delta}{2}, \sqrt{\epsilon}\right\} \\ \frac{\delta}{2} 1\{y > 0\} & \text{if } \frac{\sigma}{\tau} > \frac{\delta}{2} \text{ and } \frac{\delta}{2} \leq \sqrt{\epsilon} \\ \sqrt{\epsilon} 1\left\{y > \tau - \frac{\sigma}{\sqrt{\epsilon}}\right\} & \text{otherwise} \end{cases} \tag{131}$$

and it achieves the optimal value (26).

Proof. Recall that $\hat{\sigma}(\sigma, \tau)$ in (114) denotes the actual service supply for jobs having a service demand σ and a sojourn time τ, and $\hat{\tau}(\sigma, \tau)$ in (125) denote the actual sojourn time for such jobs. The optimization problem (130) can be written into

$$\inf_{v:(1)(14)} \text{Var}(P(t)) + \mathbb{E}[\delta U] + \mathbb{E}[\epsilon W]$$

$$= \inf_{\substack{\hat{\sigma}(\sigma,\tau) \geq \sigma \\ \hat{\tau}(\sigma,\tau) \geq \tau}} \left[\inf_{v:(1)(14)} \text{Var}(P) + \int_{(\sigma,\tau)\in S} \{\delta(\sigma - \hat{\sigma}(\sigma,\tau)) + \epsilon(\hat{\tau}(\sigma,\tau) - \tau)\} \Lambda f(\sigma,\tau) d\sigma d\tau \right]$$

$$= \inf_{\substack{\hat{\sigma}(\sigma,\tau) \geq \sigma \\ \hat{\tau}(\sigma,\tau) \geq \tau}} \int_{(\sigma,\tau)\in S} \left[\frac{\hat{\sigma}(\sigma,\tau)^2}{\hat{\tau}(\sigma,\tau)} + \delta(\sigma - \hat{\sigma}(\sigma,\tau)) + \epsilon(\hat{\tau}(\sigma,\tau) - \tau) \right] \Lambda f(\sigma,\tau) d\sigma d\tau \tag{132}$$

$$= \inf_{\substack{\hat{\sigma}(\sigma,\tau) \geq \sigma \\ \hat{\tau}(\sigma,\tau) \geq \tau}} \int_{(\sigma,\tau)\in S} C(\sigma, \hat{\tau}) \Lambda f(\sigma,\tau) d\sigma d\tau,$$

where $C(\sigma, \hat{\tau})$ is defined to be

$$C(\sigma, \hat{\tau}) := \frac{\hat{\sigma}(\sigma, \tau)^2}{\hat{\tau}(\sigma, \tau)} + \delta(\sigma - \hat{\sigma}(\sigma, \tau)) + \epsilon(\hat{\tau}(\sigma, \tau) - \tau)$$

$$= \begin{cases} \dfrac{\sigma^2}{\tau} & \text{if } \hat{\tau} = \tau \text{ and } \dfrac{\sigma}{\tau} \leq \dfrac{\delta}{2} \\[2mm] \delta\left(\sigma - \dfrac{\delta\tau}{4}\right) & \text{if } \hat{\tau} = \tau \text{ and } \dfrac{\sigma}{\tau} > \dfrac{\delta}{2} \\[2mm] \dfrac{\sigma^2}{\hat{\tau}} + \epsilon(\hat{\tau} - \tau) & \text{if } \hat{\tau} > \tau \text{ and } \dfrac{\sigma}{\hat{\tau}} \leq \dfrac{\delta}{2} \\[2mm] \delta\left(\sigma - \dfrac{\delta\hat{\tau}}{4}\right) + \epsilon(\hat{\tau} - \tau) & \text{if } \hat{\tau} > \tau \text{ and } \dfrac{\sigma}{\hat{\tau}} > \dfrac{\delta}{2} \end{cases}.$$

Relation (132) holds because $\inf_{v:(1)(14)} \mathrm{Var}(P)$ is attained by

$$v(\sigma, \tau, y) = \frac{\hat{\sigma}(\sigma, \tau)}{\hat{\tau}(\sigma, \tau)}.$$

The optimal $\hat{\sigma}^*(\sigma, \tau)$ and $\hat{\tau}^*(\sigma, \tau)$ is the point-wise maximum of the integrand of (132).

To derive a closed form expression for $\hat{\sigma}^*(\sigma, \tau)$ and $\hat{\tau}^*(\sigma, \tau)$, we first show that in the case of $\delta^2/4 \leq \epsilon$, we have $\hat{\tau}^*(\sigma, \tau) = \tau$. Suppose not and $\hat{\tau}(\sigma, \tau) = \hat{\tau} \geq \tau$. Then, if $\sigma \leq \delta\tau/2$, we have

$$C(\sigma, \hat{\tau}) - C(\sigma, \tau) = \frac{\sigma^2}{\hat{\tau}} + \epsilon(\hat{\tau} - \tau) - \frac{\sigma^2}{\tau}$$

$$= (\hat{\tau} - \tau)\left(\epsilon - \frac{\sigma^2}{\tau\hat{\tau}}\right)$$

$$\geq (\hat{\tau} - \tau)\left\{\epsilon - \left(\frac{\delta\tau}{2}\right)^2 \frac{1}{\tau\hat{\tau}}\right\} \tag{133}$$

$$\geq (\hat{\tau} - \tau)\left\{\epsilon - \frac{\delta^2}{4}\right\} \tag{134}$$

$$\geq 0, \tag{135}$$

where (133) is due to $\sigma \leq \delta\tau/2$; (134) is due to $\hat{\tau} > \tau$; and (135) is due to $\delta^2/4 \leq \epsilon$. When $\sigma \in (\delta\tau/2, \delta\hat{\tau}/2]$, we have

$$C(\sigma, \hat{\tau}) - C(\sigma, \tau) = \frac{\sigma^2}{\hat{\tau}} + \epsilon(\hat{\tau} - \tau) - \delta\left(\sigma - \frac{\delta\tau}{4}\right)$$

$$\geq \epsilon(\hat{\tau} - \tau) + \left(\frac{\delta\tau}{2}\right)^2 \frac{1}{\hat{\tau}} - \delta\frac{\delta\hat{\tau}}{2} + \frac{\delta^2\hat{\tau}}{4} \tag{136}$$

$$\geq \epsilon(\hat{\tau} - \tau) + \frac{1}{2}\delta^2(\tau - \hat{\tau}) \tag{137}$$

$$= (\hat{\tau} - \tau)\left\{\epsilon - \frac{\delta^2}{4}\right\}$$

$$\geq 0, \tag{138}$$

where (136) is due to $\sigma \leq \delta\tau/2$; (137) is due to $\hat{\tau} > \tau$; and (138) is due to $\delta^2/4 \leq \epsilon$. When $\sigma > \delta\hat{\tau}/2$, we have

$$C(\sigma, \hat{\tau}) - C(\sigma, \tau) = \delta\left(\sigma - \frac{\delta\hat{\tau}}{4}\right) + \epsilon(\hat{\tau} - \tau) - \delta\left(\sigma - \frac{\delta\tau}{4}\right)$$

$$= (\hat{\tau} - \tau)\left(\epsilon - \frac{\delta^2}{4}\right)$$

$$\geq 0 \tag{139}$$

where (139) is due to $\delta^2/4 \leq \epsilon$. Since (135), (138), and (139) contradict the supposition that $\hat{\tau}(\sigma, \tau) = \hat{\tau} > \tau$ is optimal, we have $\hat{\tau}^*(\sigma, \tau) = \tau$. Then, given $\hat{\tau}^*(\sigma, \tau) = \tau$, the optimal $\hat{\sigma}^*(\sigma, \tau)$ follows from Lemma 7. In a similar manner, we can show that, in the case of $\delta^2/4 > \epsilon$, the optimal service supply is $\hat{\sigma}^*(\sigma, \tau) = \sigma$. Then, given $\hat{\sigma}^*(\sigma, \tau) = \sigma$, the optimal $\tau^*(\sigma, \tau)$ follows from Lemma 8. Finally, combining above, we obtain (131) as the closed-form solution of (130). Q.E.D.

Theorem 4 is an immediate consequence of Lemma 9. To see it, recall that the optimal value of problem (130) lower-bounds that of problem (9). Moreover, scheduler (25) of the form (4) can produce identical service rates to (131), so it is also optimal for problem (9).

Appendix F: Proof of Lemma 4

To solve $\inf_{c:(29)} L(c;\gamma)$, we first observe that

$$\inf_{c:(29)} L(c;\gamma) = \inf_{c:(29)} \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathrm{Var}(P(t)) + \gamma(\mathrm{Var}(X(t)) - D)dt \tag{140}$$

$$\geq \inf_{c:(29)} \lim_{T\to\infty} \inf_{c:(29)} \frac{1}{T}\int_0^T \mathrm{Var}(P(t)) + \gamma(\mathrm{Var}(X(t)) - D)dt \tag{141}$$

$$= \lim_{T\to\infty} \inf_{c:(29)} \frac{1}{T}\int_0^T \mathrm{Var}(P(t)) + \gamma(\mathrm{Var}(X(t)) - D)dt, \tag{142}$$

where equality (140) holds by the definition of $L(c;\gamma)$, inequality (141) holds because $(1/T)\int_0^T \mathrm{Var}(P(t)) + \gamma(\mathrm{Var}(X(t)) - D)dt$ is always less than $(1/T)\inf_{c:(29)}\int_0^T \mathrm{Var}(P(t)) + \gamma(\mathrm{Var}(X(t)) - D)dt$.

Now we consider representing the integral of (142) as the sum of $\mathbb{E}[(P(t_n) - \bar{P})^2 + \gamma(X(t_n) - \bar{X})^2]$ at discrete points in time, where $\{t_n\}$ have a fixed sampling interval $h = t_{n+1} - t_n, \forall n \in \mathbb{Z}_+$. So, the dynamics of $X(t_n)$ satisfies

$$X(t_{n+1}) = X(t_n) + A(t_n, h) - hP(t_n), \tag{143}$$

where $A(t_n, h)$ is the demand added to X due to new arrivals in the time interval $[t_n, t_{n+1})$ (the total demands of jobs arriving at this interval), $hP(t_n)$ is the total service provided during $[t_n, t_{n+1})$. Here, the service policy c is assumed to produce constant values during each sampling intervals, *i.e.* $c(k, t_1, A_t) = c(k, t_2, A_t)$ for any $t_1, t_2 \in [t_n, t_{n+1})$,[19] so the service capacity takes the constant value $P(t_n)$ during this interval.

Let $L_{h,N}(u;\gamma)$ is defined by[20]

$$L_{h,N}(c;\gamma) := \mathbb{E}\left[\gamma(X(t_N) - \bar{X})^2\right] + \sum_{k=0}^{N-1} \mathbb{E}\left[(P(t_k) - \bar{P})^2 + \gamma(X(t_k) - \bar{X})^2\right].$$

Observe that (142) satisfies

$$\lim_{T\to\infty} \inf_{c:(29)} \frac{1}{T}\int_0^T \mathbb{E}[(P(t) - \bar{P})^2 + \gamma((X(t) - \bar{X})^2 - D)]dt$$

$$= \lim_{T\to\infty} \inf_{c:(29)} \lim_{h\to0} \frac{1}{T}L_{h,\lceil T/h\rceil}(c;\gamma)h - \gamma D$$

$$= \lim_{T\to\infty} \lim_{h\to0} \inf_{c:(29)} \frac{1}{T}L_{h,\lceil T/h\rceil}(c;\gamma)h - \gamma D. \tag{144}$$

To solve (144), we first consider the cost-to-go $J_n(X(t_n))$ for some $h > 0$ and $N \in \mathbb{Z}_+$, *i.e.*

$$J_n(X(t_n)) := \mathbb{E}\left[\gamma(X(t_N) - \bar{X})^2\right] + \sum_{k=n}^{N-1} \mathbb{E}\left[(P(t_k) - \bar{P})^2 + \gamma(X(t_k) - \bar{X})^2\right]. \tag{145}$$

Using mathematical induction, we show below that, at the optimal solution c^*, the cost-to-go takes the form

$$J_n(X(t_n)) = \mathbb{E}\left[p_n(X(t_n) - \bar{X})^2\right] + \sum_{k=n}^{N-1} \mathbb{E}[p_{k+1}(A(t_n, h) - \bar{A}_h)^2], \tag{146}$$

where $\{p_k\}$ satisfies the Riccati difference equation

$$p_k = p_{k+1} - \frac{h^2 p_{k+1}^2}{h^2 p_{k+1} + 1} + \gamma, \qquad\qquad p_N = \gamma. \tag{147}$$

---

[19] As the sampling interval goes to zero, c can realize any continuous function $c(k, t, A_t)$ of t.

[20] Here, $\bar{X}$ and $\bar{P}$ is the expected value of $X(t)$ and $P(t)$, which is not related with the sampling interval. It shall also be noted that the subsequent analysis assumes a fixed sampling interval h, which is sufficiently small.

First, condition (146) holds for $n = N$ by the construction of (145). Second, assume that condition (146) holds for $n+1$. Let $\bar{A}_h$ be the stationary mean of $A(t_n, h)$. Recall from (145) that $J_n(X(t_n))$ is the sum of term n to term N. Thus, it can be decomposed into the term of n and the sum of $n+1$ term to N term, which is $J_{n+1}(X(t_{n+1}))$. So, we have

$$J_n(X(t_n))$$

$$= \inf_{P(t_n)} \mathbb{E}[(P(t_n) - \bar{P})^2 + \gamma(X(t_n) - \bar{X})^2 + J_{n+1}(X(t_{n+1}))] \tag{148}$$

$$= \inf_{P(t_n)} \mathbb{E}[(P(t_n) - \bar{P})^2 + \gamma(X(t_n) - \bar{X})^2 + J_{n+1}(X(t_n) + A(t_n, h) - hP(t_n)] \tag{149}$$

$$= \inf_{P(t_n)} \mathbb{E}[(P(t_n) - \bar{P})^2 + \gamma(X(t_n) - \bar{X})^2 + J_{n+1}(X(t_n) + (A(t_n, h) - \bar{A}_h) - h(P(t_n - \bar{P}))] \tag{150}$$

$$= \inf_{P(t_n)} \mathbb{E}[(P(t_n) - \bar{P})^2 + \gamma(X(t_n) - \bar{X})^2 + p_{n+1}(X(t_n) + (A(t_n, h) - \bar{A}_h) - h(P(t_n) - \bar{P}))^2] \tag{151}$$

$$+ \sum_{k=n+1}^{N-1} \mathbb{E}[p_{k+1}(A(t_k, h) - \bar{A}_h)^2]$$

where (149) uses relation (143); (150) relies on $\bar{A}_h = h\bar{P}$ from Brumelle's formula; (151) uses the induction hypothesis that the cost-to-go at $n+1$ satisfies (146). Note that

$$\mathbb{E}[(A(t_n, h) - \bar{A}_h)X(t_n)] = \mathbb{E}[A(t_n, h) - \bar{A}_h]\mathbb{E}[X(t_n)] = 0, \tag{152}$$

where the first equality holds because future arrivals in interval $[t_n, t_{n+1})$ does not depend on past arrivals in interval $[0, t_n)$, and the second equality is due to $\mathbb{E}[A(t_n, h) - \bar{A}_h] = 0$. Expanding the last quadratic term in (148) and applying $\mathbb{E}[(A(t_n, h) - \bar{A}_h)X(t_n)] = 0$, we can rewrite (152) into

$$J_n(X(t_n)) = (p_{n+1} + \gamma)(X(t_n) - \bar{X})^2 + \sum_{k=n}^{N} p_{k+1}\mathbb{E}(A(t_k, h) - \bar{A}_h)^2]$$

$$+ \inf_{P(t_n)} \{(1 + h^2 p_{n+1})(P(t_n) - \bar{P})^2 - 2h\gamma p_{n+1}(X(t_n) - \bar{X})(P(t_n) - \bar{P})]. \tag{153}$$

The minimum value of (153) is attained by

$$P(t_n, h) - \bar{P}_h = \frac{hp_n}{1 + h^2 p_n}(X(t_n) - \bar{X}), \tag{154}$$

and the optimal cost-to-go becomes (146), where p_n is defined by (147). As $N \to \infty$, p_k converges to a unique positive scalar

$$p := \lim_{N \to \infty} p_k = \frac{h^2\gamma + h\sqrt{\gamma}\sqrt{h^2\gamma + 4}}{2h^2}, \tag{155}$$

which is also a fixed point of (147) [50]. Taking the limit of $N \to \infty$ and $h \to 0$ for (154) and (155), the infimum of (144) is attained when

$$P(t) - \bar{P} = \sqrt{\gamma}\,(X(t) - \bar{X}).$$

Finally, the infimum value of (144) is computed as

$$\lim_{T \to \infty} \inf_{c:(29)} \frac{1}{T} \int_0^T \mathrm{Var}(P(t)) + \gamma(\mathrm{Var}(X(t)) - D)dt$$

$$= \lim_{T \to \infty} \lim_{h \to 0} \frac{h}{T} \sum_{k=0}^{N-1} \mathbb{E}[p_{k+1}(A(t_k, h) - \bar{A}_h)^2] - \gamma D \tag{156}$$

$$= \sqrt{\gamma}\Lambda\mathbb{E}[\sigma^2] - \gamma D \tag{157}$$

$$= \sqrt{\gamma}\Lambda\mathbb{E}[\sigma^2] - \gamma\mathrm{Var}(X). \tag{158}$$

where equality (156) is due to (146); and equality (157) is derived from (155), and equality (158) is derived from the complementary slackness condition of

$$\gamma(\mathrm{Var}(X) - D) = 0.$$

Appendix G: Proof of Corollary 2

Recall from Lemma 3 that $X(t)$ is the total remaining demands of jobs arriving before t. For any time interval $h > 0$, $X(t)$ satisfies the following dynamics:

$$X(t+h) = X(t) + A(t,h) - P(t,h),$$

where $A(t,h)$ is the total demand of jobs arriving during time interval $[t, t+h)$, and $P(t,h)$ is the total amount served during this interval, *i.e.*

$$A(t,h) := \sum_{\{k \in \mathcal{V}: a_k \in [t,t+h)\}} \sigma_k,$$

$$P(t,h) := \int_t^{t+h} P(\tau) d\tau.$$

let $D_t = \{k \in \mathcal{V} : a_k + \tau_k \leq t\}$ be the set of jobs that departs by time t. As no job receives more service than its demand, $X(t)$ is bounded from above by

$$\begin{aligned} X(t) &= \sum_{k \in A_t} \sigma_k - \int_{\tau \geq t} P(\tau) d\tau \\ &\leq \sum_{k \in A_t} \sigma_k - \sum_{i \in D_t} \sigma_k \\ &\leq \sum_{k \in A_t \setminus D_t} \sigma_k \end{aligned} \tag{159}$$

where $D_t = \{k \in \mathcal{V} : a_k + \tau_k \leq t\}$ is the set of jobs that departs by time t. From (159) and $X(t) \geq 0$, the variance of $X(t)$ is upper-bounded by

$$\text{Var}(X(t)) \leq \mathbb{E}[X(t)^2] \tag{160}$$

$$\leq \mathbb{E}\left[\left(\sum_{k \in A_t \setminus D_t} \sigma_k\right)^2\right]$$

$$= \text{Var}\left(\sum_{k \in A_t \setminus D_t} \sigma_k\right) + \mathbb{E}\left[\sum_{k \in A_t \setminus D_t} \sigma_k\right]^2$$

$$= \int_{(\sigma,\tau) \in S} \tau \sigma^2 \Lambda f(\sigma,\tau) d\sigma d\tau + \left(\int_{(\sigma,\tau) \in S} \tau \sigma \Lambda f(\sigma,\tau) d\sigma d\tau\right)^2$$

$$= \Lambda \mathbb{E}\left(\tau\sigma^2\right) + (\Lambda \mathbb{E}[\tau\sigma])^2 \tag{161}$$

Applying $D = \Lambda \mathbb{E}\left(\tau\sigma^2\right) + (\Lambda \mathbb{E}[\tau\sigma])^2$ to Lemma 3, we obtain (39).

Appendix H: Additional performance bound.

Lemma 3 characterizes the tradeoff between achieving a small variance of $X(t)$ and achieving a small variance of $P(t)$. Plugging in Exact Scheduling's stationary variance of X,

$$\text{Var}(X) = \Lambda \mathbb{E}\left[\frac{1}{3}\sigma^2\tau\right],$$

we obtain a performance bound for Exact Scheduling (13).

Corollary 7. *Let* $\text{Var}(P)$ *be the stationary variance of* $P(t)$ *that is attained by Exact Scheduling* (13). *Let* $\text{Var}(P^\dagger)$ *be the minimum stationary variance attainable by any centralized algorithm* (29) *with the same level of* $\text{Var}(X)$ *as Exact Scheduling. Then, the following condition holds:*

$$\text{Var}(P) \leq \frac{4}{3} \frac{\mathbb{E}[\sigma^2/\tau]\mathbb{E}[\sigma^2\tau]}{\mathbb{E}[\sigma^2]^2} \text{Var}(P^\dagger),$$

where the expectations on the right hand side are taken over the arrival distribution.

In the setting of soft service requirements, Generalized Exact Scheduling attains

$$\mathrm{Var}(X) = \Lambda\mathbb{E}\left[\frac{\sigma^2\tau}{3}\mathbf{1}\left\{\frac{\sigma}{\tau} \le \min\left\{\frac{\delta}{2}, \sqrt{\epsilon}\right\}\right\}\right] + \Lambda\mathbb{E}\left[\left(\frac{\delta^2\tau^3}{12} - \frac{1}{2}\delta\sigma\tau^2 + \sigma^2\tau\right)\mathbf{1}\left\{\frac{\sigma}{\tau} > \frac{\delta}{2} \ge \sqrt{\epsilon}\right\}\right]$$
$$+ \Lambda\mathbb{E}\left[\left(\frac{\sigma^3}{3\sqrt{\epsilon}}\right)\mathbf{1}\left\{\frac{\sigma}{\tau} > \sqrt{\epsilon} > \frac{\delta}{2}\right\}\right].$$

Combining above and Lemma 3, we obtain the following corollary.

Corollary 8. *Let* $\mathrm{Var}(P)$ *be the stationary variance of* $P(t)$ *that is attained by Generalized Exact Scheduling* (25). *Let* $\mathrm{Var}(P^*)$ *be the minimum stationary variance attainable by any centralized algorithm of the form* (29) *with the same level of* $\mathrm{Var}(X)$ *as Generalized Exact Scheduling. Then, the following condition holds:*

$$\mathrm{Var}(P) \le \frac{\alpha\beta}{\mathbb{E}[\sigma^2]^2}\mathrm{Var}(P^*),$$

where

$$\alpha = \mathbb{E}\left[\frac{\sigma^2}{\tau}\mathbf{1}\left\{\frac{\sigma}{\tau} \le \min\left\{\frac{\delta}{2}, \sqrt{\epsilon}\right\}\right\} + \delta\left(\sqrt{\epsilon} - \frac{\delta\tau}{4}\right)\mathbf{1}\left\{\frac{\sigma}{\tau} > \frac{\delta}{2} \ge \sqrt{\epsilon}\right\} + \left(2\sqrt{\epsilon}\sigma - \epsilon\tau\right)\mathbf{1}\left\{\frac{\sigma}{\tau} > \sqrt{\epsilon} > \frac{\delta}{2}\right\}\right]$$

$$\beta = \mathbb{E}\left[\frac{\sigma^2\tau}{3}\mathbf{1}\left\{\frac{\sigma}{\tau} \le \min\left\{\frac{\delta}{2}, \sqrt{\epsilon}\right\}\right\} + \left(\frac{\delta^2\tau^3}{12} - \frac{1}{2}\delta\sigma\tau^2 + \sigma^2\tau\right)\mathbf{1}\left\{\frac{\sigma}{\tau} > \frac{\delta}{2} \ge \sqrt{\epsilon}\right\} + \left(\frac{\sigma^3}{3\sqrt{\epsilon}}\right)\mathbf{1}\left\{\frac{\sigma}{\tau} > \sqrt{\epsilon} > \frac{\delta}{2}\right\}\right].$$

Corollary 8 bounds the ratio of $\mathrm{Var}(P)$ achievable by Generalized Exact Scheduling (the optimal distributed algorithm) to $\mathrm{Var}(P^*)$ achievable by any centralized algorithms. Here, the optimal distributed algorithm is subject to soft service constraints, while the optimal centralized algorithm is subject to the same $\mathrm{Var}(X)$ with Generalized Exact Scheduling.

Appendix I: Additional numerical results

Section 4.1 shows the empirical performance of different algorithms for typical cases. In this section, we provide more detailed experimental data to support the results in Section 4.1. Figure 10 compares how Exact Scheduling and Offline Optimal schedule jobs in two instances: one instance in which Exact Scheduling performed competitively, and another instance in which Exact Scheduling performed poorly. Figure 11 provide a more comprehensive view of Figure 5 by comparing the algorithms' performance for the arrival distribution of a broader range of parameters.

(a) Example case when Exact Scheduling performed competitively to Offline Optimal
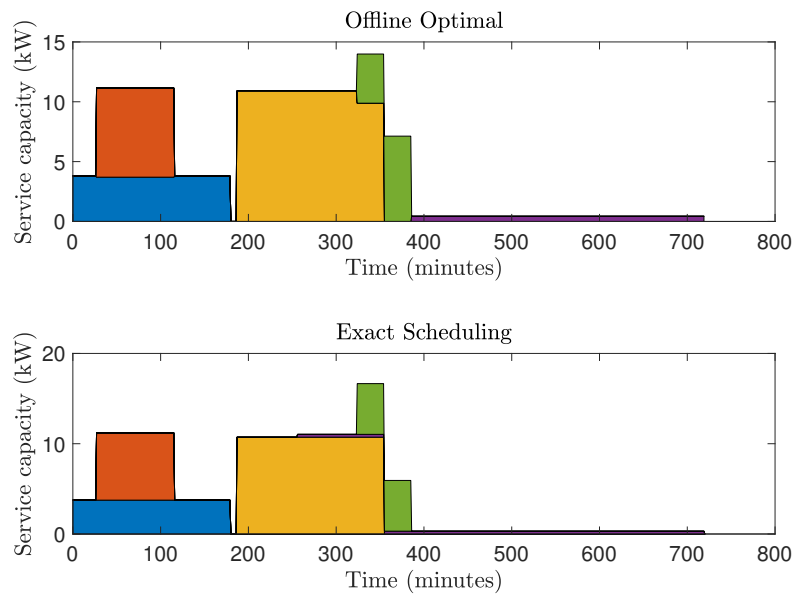

(b) Example case when Exact Scheduling performed poorly to Offline Optimal
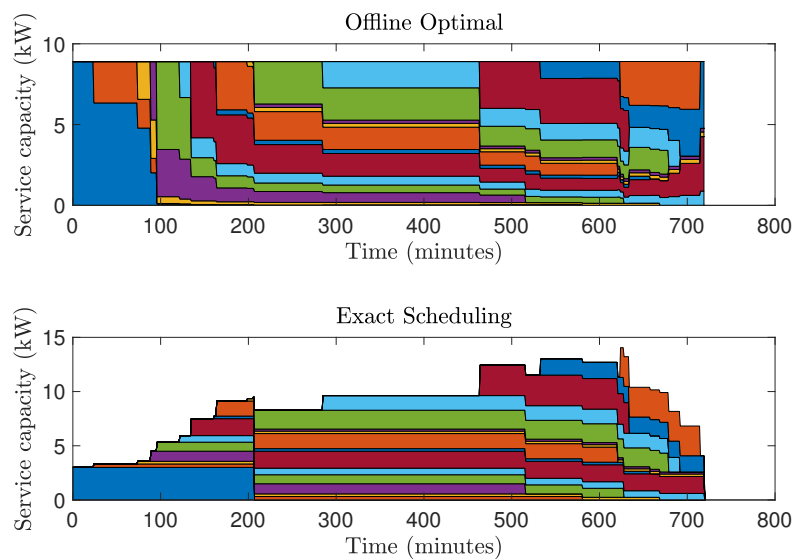
Figure 10　**Example cases when Exact Scheduling performs competitively or poorly in comparison to Offline Optimal. Each colored region represents the service rate for one job over its sojourn time, and the height of the colored region (by any color) shows the sum of service rate, i.e. the service capacity, at each time. The sum of all service rates at time t is the service capacity $P(t)$. The top plot (a) shows a case when Exact Scheduling performs competitively to Offline Optimal, while the bottom plot (b) shows a case when the Offline Optimal has much better performance than Exact Scheduling.**

(a) Performance in synthetic data generated from arrival distribution I.
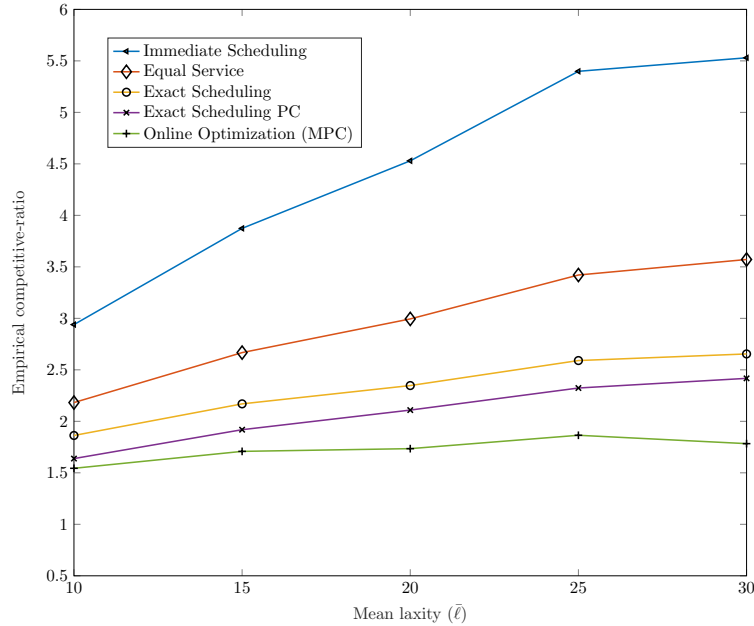
(b) Performance in synthetic data generated from arrival distribution II.
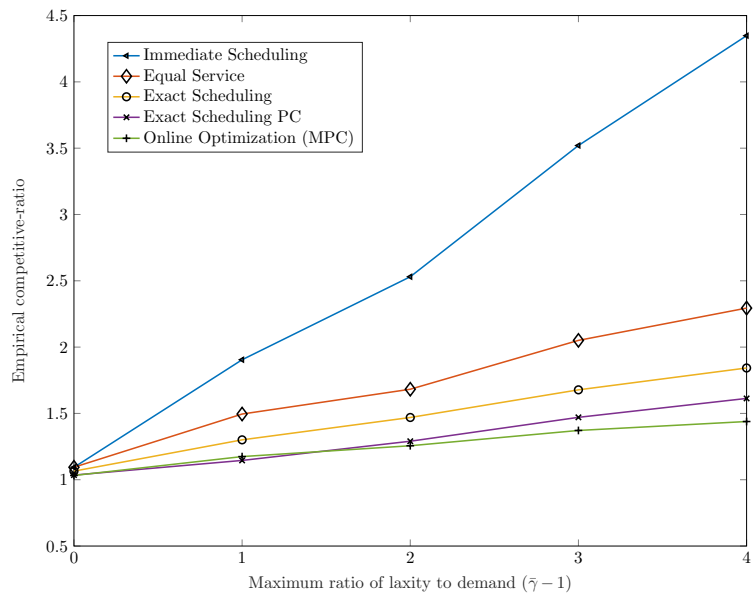
Figure 11   **Performance comparison of algorithms under strict service (demand and deadline) constraints for varying parameters of arrival distribution. The ratio of each algorithm's empirical variance to the Offline Optimal is averaged over all scheduling instances. The number of instances averaged are** 500 **for both plots. In plot (a), the mean laxity refers to parameter** $\bar{\ell}$ **in distribution I, and the empirical competitive-ratio for** $\bar{\ell} = 25$ **is shown in Figure 5b. In plot (b), the maximum ratio of laxity to demand refers to** $\bar{\gamma} - 1$ **in distribution II, and the empirical competitive-ratio (the realized cost** (6) **divided the cost of the optimal offline algorithm) for varying** $\bar{\gamma} - 1 = 1$ **is shown in Figure 5c.**