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 infrastructure 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 distribution. 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 demand as it changes over time and, 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.

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 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 below 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, our goal is to derive general optimality results that hold beyond the heavy-traffic regime. Further, we seek to design optimal *distributed algorithms*, which only use local information about each job to decide the desirable service rate. Those algorithms 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 this constraints in 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. Further, we provide a novel competitive-ratio-like bounds 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 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 competitive-ratio-like 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 an extensive literature that studies the design and analysis of deadline scheduling algorithms (see [3, 16–18] and references therein). Examples of classic scheduling algorithms include Earliest Deadline First [7–9,19–21] and Least Laxity First [19], among others [22,23]. Beyond these classic algorithms, more modern algorithms simultaneously perform admission control and service rate control to exploit the flexibility arising from soft demand or deadline requirements, *e.g.* [24–26].

The trade-offs between service quality and costs associated with variability have become a focus only recently [12,27,28], motivated by applications such as cloud computing and power distribution systems. In the context of cloud computing, algorithms have been proposed to control the variability of power usage in data centers using deferrable jobs (see [29–40] 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 [41–45] and references therein).

Interesting optimality results have been obtained in some limited settings, such as deterministic worst-case settings [6, 14], single server systems [7–9], and/or heavy traffic settings [10, 11]. For example, in heavy traffic settings, the dynamic behavior of discrete queueing systems can be approximated by a continuous-state process involving Brownian motion, for which there exist established tools to optimize [10, 11]. On the other hand, optimizing queueing systems without continuous-state approximations remains to be a hard problem. Particularly, the problem of designing *optimal* algorithms that minimize service capacity variability while achieving high service quality has remained open. Solving this problem is a challenging task due to the heterogeneity of jobs (diversity in demands and deadlines) and the size of the state and decision space (of possible configurations on existing job profiles and the set of feasible scheduling policies).

However, the problem of designing *optimal* algorithms that minimize service capacity variability while achieving high service quality has remained open. 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 particular, the only optimality results that have been obtained to this point are in niche settings such as a static single server system [7–9] and deterministic worst-case settings [6,14].

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 $S \times C$ [17]. 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 $\Lambda(a,\sigma,\tau) = \tilde{\Lambda}(a)f_a(\sigma,\tau)$. Intuitively, the intensity function is the average rate at which jobs with service requirement (σ,τ) arrive at time a. When both $\tilde{\Lambda}(a) \equiv \tilde{\Lambda}$ 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 and the sojourn time τ has finite first and second moments, S is bounded, and $S \subset \{(\sigma,\tau) : \tau \geq \sigma \text{ and } \sigma \geq 0\}$.[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 $\bar{r}$, and without loss of generality, we assume that $\bar{r} = 1$, *i.e.*

$$r_k(t) \in [0,\ 1]. \tag{1}$$

To meet the demand requirements, the service rate must satisfy

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

To meet the deadline requirements, it also need to satisfy

$$r_k(t) \leq \mathbf{1}\{a_k \leq t < a_k + \tau_k\}, \qquad k \in \mathcal{V}. \tag{3}$$

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

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

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

[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.

[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.

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 unsatisfied 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 $\hat{\tau}_k = \max\{t - a_k : r_k(t) > 0\}$ 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 help reduce cost. Thus, our design objective is to reduce the variance and mean square in service capacity for the settings with strict or soft service constraints. Specifically, we consider the optimization problem

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

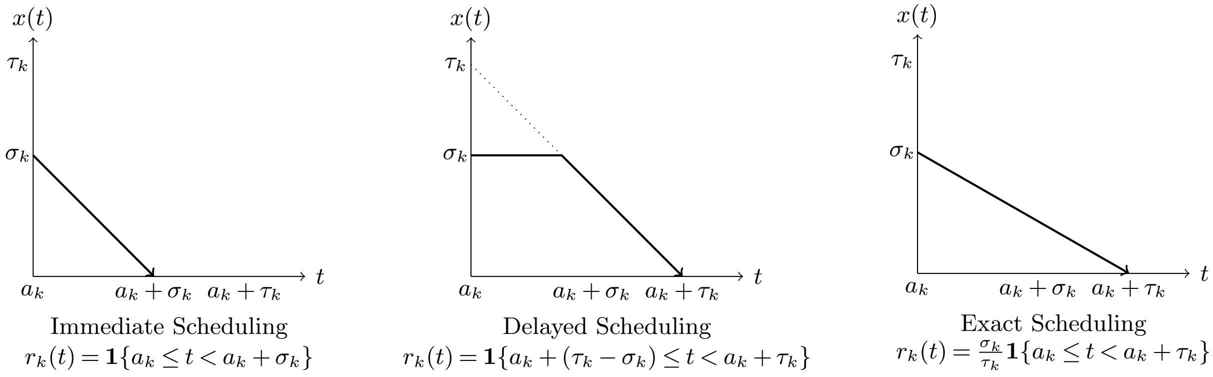
where the first term of the integrand quantifies the service capacity stability, and the second term quantifies the service capacity predictability. The coefficient $\alpha, \beta (\geq 0)$ balances the predictability and stability of $P(t)$, and the objective function reduces to the time average of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

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

which balances service capacity variance with the penalties for not meeting the demands and/or deadlines of some jobs.

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 results 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.



Immediate Scheduling
$r_k(t) = \mathbf{1}\{a_k \le t < a_k + \sigma_k\}$

Delayed Scheduling
$r_k(t) = \mathbf{1}\{a_k + (\tau_k - \sigma_k) \le t < a_k + \tau_k\}$

Exact Scheduling
$r_k(t) = \frac{\sigma_k}{\tau_k}\mathbf{1}\{a_k \le t < a_k + \tau_k\}$

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.**

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 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 a 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% [46]. 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 [6,14,15,19,43,47,48]. 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 [48]. 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 3.3). 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 the settings when 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 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. 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 [17]. 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 [49, 50]. 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. 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.

[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.

Proposition 1. *Consider a service system with a 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*

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

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 [51, Chapter 3, eq. (3.2.1)].

As the stationary mean of $P(t)$ does not depend on the specific form of the policy (4), we consider minimizing the stationary variance of $P(t)$ under strict service constraints, soft demand constraints, soft deadline constraints, and soft demand and deadline constraints. In the case of *strict demand 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 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 penality 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

[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.

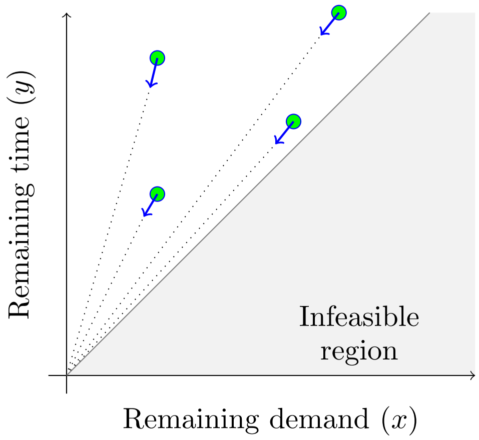


Figure 2 **Exact scheduling depicted in the space of remaining demand x and remaining time y.**

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:

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

3.2. Strict 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 $\text{Var}(P)$. To do so, it is worth noticing from Lemma 2 that service rates contribute to the service capacity variance in a quadratic form. 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 2). 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, 52], no work has shown its optimality conditions. 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]

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

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

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 an 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

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

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).

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

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

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

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

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 $\mathrm{Var}(P)$ can be computed by integrating along the trajectory of a job over its distribution $\Lambda f(\sigma, \tau)$ [17].[8] In particular, the following relation holds.

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

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

[8] This is a restatement of Brumelle's formula [53] from queueing theory for systems with infinitely many servers with time-varying rates.

Lemma 2 can be obtained from the 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

$$\begin{aligned}\mathrm{Var}(P) &= \int_{(\sigma,\tau)\in S}\int_0^\tau v(\sigma,\tau,y)^2 \Lambda f(\sigma,\tau)dy d\sigma d\tau && (18)\\ &= \int_{(\sigma,\tau)\in S}\left\{\int_0^\tau v(\sigma,\tau,y)^2 dy\right\}\Lambda f(\sigma,\tau)d\sigma d\tau && (19)\\ &\geq \int_{(\sigma,\tau)\in S}\left\{\frac{\sigma^2}{\tau}\right\}\Lambda f(\sigma,\tau)d\sigma d\tau. && (20)\end{aligned}$$

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 considering accounting for job arrival times by consider 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 penalty δ for each unit of unsatisfied demands. The resulting optimal algorithm is a variation of Exact Scheduling with an additional rate upper-bound:

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

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*

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

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 $\mathbb{E}\left[\sigma^2/\tau\right]$ 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 $\mathbb{E}\left[\sigma^2/\tau\right]$ to $\mathbb{E}\left[\delta\left(\sigma - (\delta\tau/4)\right)\right]$. And the portion of such jobs are given by $\mathbb{E}[\mathbf{1}\left\{\sigma/\tau > \delta/2\right\}]$. 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) often leads to an NP-hard problem [54,55], by taking a probabilistic approach aimed at finding the best remaining job distribution, we obtain the optimal algorithm in closed-form below. The resulting optimal algorithm is a variation of Exact Scheduling with deadline extensions:

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

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

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

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

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 are given by $\mathbb{E}\left[\mathbf{1}\{\sigma/\tau > \sqrt{\epsilon}\}\right]$. Moreover, service capacity variance and penalties for deadline extension is 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 reject 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:

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

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 3 as the white, light gray, and dark gray 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 Section 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*

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

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*

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

[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.

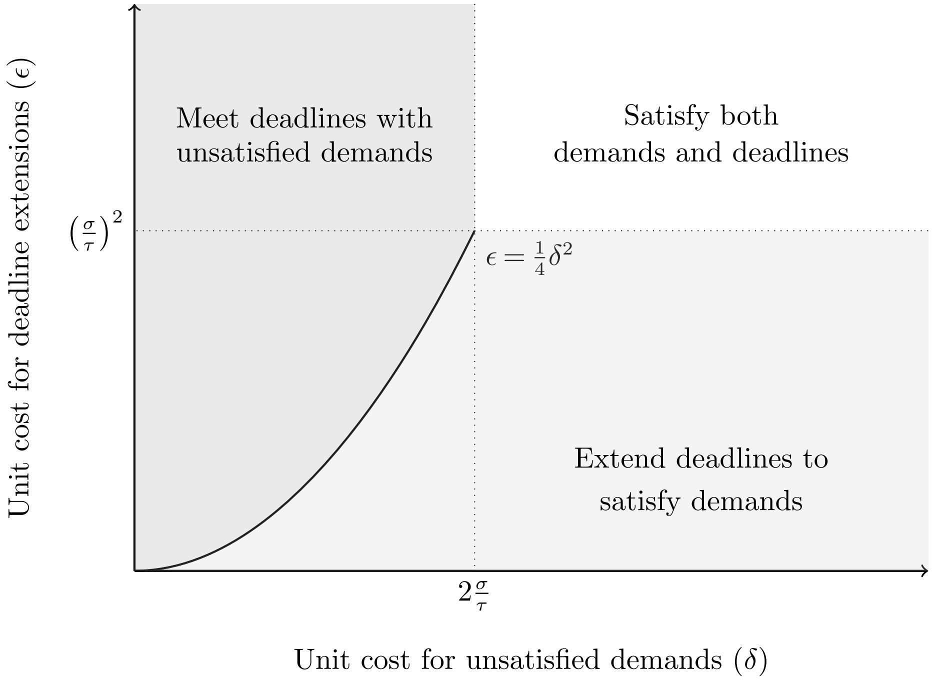

Unit cost for deadline extensions (ϵ)
$\left(\frac{\sigma}{\tau}\right)^2$
Meet deadlines with unsatisfied demands
Satisfy both demands and deadlines
$\epsilon = \frac{1}{4}\delta^2$
Extend deadlines to satisfy demands
$2\frac{\sigma}{\tau}$
Unit cost for unsatisfied demands (δ)

Figure 3 **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 (dark gray region), or satisfy the demand by extending the deadline (light gray region).**

Corollary 1 is an immediate consequence of Theorem 4.

4. Performance degradation inherent to availability in job information

Given our focus on distributed algorithms, we should investigate how much performance degrades in comparison to centralized algorithms. This investigation also includes a practically important question: if there is any middle ground between centralized and distributed algorithms in which scalability and close-to-centralized performance can be achieved simultaneously. Moreover, another practically relevant question is: what can be done if the information on the service requirement (demands and/or deadlines) are missing, and how much does the performance degrade due to the missing information? In this section, we answer these questions using both 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). 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 algorithm, we compare its performance and existing scheduling algorithms in an electric vehicle charging testbed and using synthetic data of varying arrival distribution.

(a) Job profiles

	Demand σ_k (kW × minutes)	Sojourn time τ_k (minutes)
Mean	$5.1 \cdot 10^2$	$4.5 \cdot 10^3$
Variance	$3.1 \cdot 10^5$	$7.7 \cdot 10^6$

(b) Instance profiles

	Total demand $\sum_{k \in \mathcal{V}} \sigma_k$ (kW × minutes)	Time horizon T (minutes)	Number of jobs $\lvert\mathcal{V}\rvert$
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 1 Statistics of the electric vehicle charging instances at the testbed [15].

4.1.1. System and data We employed a trace-driven simulation on real data from an electric vehicle charging testbed [15], and a synthetic data set randomly drawn from a set of arrival distribution with varying parameters. The real data contain 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 1. The synthetic data are generated from the following set of arrival distribution. The 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 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 tested We compared a few standard schedulers and Generalized Exact Scheduling. The schedulers range from optimal offline policy and online fully-centralized policies to online partially-centralized policies and online distributed policies. The detail of these schedulers is defined below.

Offline optimal algorithms (centralized).

To understand the best possible performance, we compare with the optimal offline algorithms. 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 the service rate at time t is allowed to use the information of all future arrivals. The optimal offline policy 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 between the cost of the algorithm and that of Offline Optimal (with a slight abuse of notation,[10] we denote this ratio as the *empirical competitive-ratio*). This quantify is used in Figure 4, 5, 10 to evaluate the performance of different algorithms under varying arrival distribution.

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 $A_t = \{(a_k, \sigma_k, \tau_k, x_k(t), y_k(t)) : a_k \leq t\}$ is the set that contains the information of jobs arriving no later than t, and $c(k, t. \cdot)$ is a deterministic mapping from A_t to a service rate $r_k(t)$. When deciding the service rate at each time, the policy can use the information of jobs that arrived before that time. We list below a few centralized online policies tested in this paper.

- Online Optimization MPC. This policy performs Model Predictive Control (MPC) on the objective function (28). Specifically, at each time t, this policy solves the optimization problem (28) with optimization variable (29) to find the service rates from t to the T, but the scheduler put only $r(t)$ into action. It recompute problem (28) to obtain the service rates from $t+1$ to the T and put only $r(t+1)$ into action. The computed service rates are optimal at t if no jobs arrive in the future but have no global optimality guarantees otherwise.
- Earliest Deadline First (EDF). This policy allocates a fixed capacity p_{EDF} to jobs in ascending order of their deadlines. Under soft demand constraints, jobs are served until their deadline. Under soft deadline constraints, jobs are served until their completion.
- Least Laxity First (LLF). Recall from (32) that $\ell_k(t) = x_k(t) - y_k(t)$ is the laxity of job k at time t. This policy allocates a fixed capacity p_{LLF} to jobs in ascending order of their laxity. Under soft demand constraints, jobs are served until their deadlines. Under soft deadline constraints, jobs are served until their completion.
- Fair Sharing (FS). This policy equally distributes a fixed capacity among jobs. Under soft demand constraints, jobs are served until their deadlines according to $r_k(t) = \min\{p_{\text{FS}}/n(t), 1\}\mathbf{1}\{y_k(t) > 0\}$, where $n(t)$ the is number of unfinished jobs at time t. Under soft deadline constraints, jobs are served to their completion according to $r_k(t) = \min\{p'_{\text{FS}}/n(t), 1\}\mathbf{1}\{x_k(t) > 0 \text{ and } y_k(t) > 0\}$. Intuitively, p_{FS} or p'_{FS} is the service capacity the system is willing to provide in their respective settings, and this capacity is shared among all unfinished jobs. Here p_{FS} and p'_{FS} are chosen to be the optimal offline values.[11]

Distributed online algorithms.

Recall from Section 3.1 that a distributed online policy takes the form (4). Below lists the centralized online policy 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 schedule jobs at its maximum rate upon arrival, *i.e.* $r_k(t) = \mathbf{1}\{x_k(t) > 0\}$ for any $k \in \mathcal{V}$.

Partially-centralized algorithms.

The centralized algorithms listed above require the scheduler to access all service requirement information (demands and deadlines) for all jobs present in the system. In contrast, the above distributed algorithms use no centralized information (*i.e.* the service rate of each job is only

[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.

[11] Since the offline optimal parameters are unknown in practice, the test results obtained here are optimistic.

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 the design of scalable algorithms, we focus on near-distributed policies that only use a limited number of global variables. Such policies take the form

$$r_k(t) = pc(x_k(t), y_k(t), z(t)) \geq 0, \qquad k \in \mathcal{V} \tag{30}$$

where $z(t)$ is a low-dimensional vector that is shared among the local schedulers for each job. The policy (30) requires much less resource in computation and communication compared with 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, (30) only evaluates a closed-form function, and $z(t)$ through to local schedulers with minimum communication resources.

We tested many algorithms, where $z(t)$ contains the service capacity, total remaining demands, total remaining time, number of jobs, and combinations of these quantities. Some has better performance than others, and we present below two of such algorithms.

- 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.*

$$pc(x_k(t), y_k(t), z(t)) = \begin{cases} \mu \dfrac{x_k(t)}{y_k(t)} & P(t-dt) < \bar{P} \\ \dfrac{x_k(t)}{y_k(t)} & \text{otherwise} \end{cases} \tag{31}$$

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

- Equal Service. This policy offers a homogeneous service rate to all unfinished jobs. Under 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. Specifically, the laxity of a job at time t is defined as the remaining time before deadline if the job is to be served at its maximum rate from time t until job completion. In this context, it can be computed as the remaining demand minus the remaining time:

$$\ell_k(t) := x_k(t) - y_k(t). \tag{32}$$

The service rate under this policy is given by $r_k(t) = c_{\mathrm{ES}}\mathbf{1}\{\ell_k(t) > 0 \text{ and } x_k(t) > 0\} + \mathbf{1}\{\ell_k(t) \leq 0 \text{ and } x_k(t) > 0\}$. Under soft demand constraints, it serves jobs at a homogeneous service rate c'_{ES} before its deadline, and the service rate is given by $r_k(t) = c'_{\mathrm{ES}}\mathbf{1}\{x_k(t) > 0 \text{ and } y_k(t) \geq 0\}$. Note that this policy may not fulfill the demand of all jobs but does satisfy all deadlines. Under soft deadline constraints, it serves jobs at a homogeneous service rate until its completion, and the service rate is given by $r_k(t) = c''_{\mathrm{ES}}\mathbf{1}\{x_k(t) > 0\}$. 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 $c_{\mathrm{ES}}, c'_{\mathrm{ES}}, c''_{\mathrm{ES}}$ are closed to $\mathbb{E}[v^*(\sigma_k, \tau_k, y_k(t))]$, where v^* is the optimal solutions for problem (15), (77), and (88).[12] Indeed, we observed this behavior empirically. Moreover, we noticed that Equal Service has robust behavior and performance to small perturbation in its rate away from these values as well.

The algorithms listed above are compared in the settings of strict service constraints in Figure 4, soft demand constraints in Figure 6, soft deadline constraints in Figure 7.

[12] Problem (77) and (88) are defined in the Appendix.

4.1.3. Fully-centralized vs partially-centralized vs distributed algorithms 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 requirement strictly. For the instances in the testbed, we observed a significant performance degradation from Offline Optimal to Online algorithms: online algorithms experience 1.5 times more cost due to the lack of future information. However, among online algorithms, Exact Scheduling only degrades from Online Optimization (MPC) by an average of 20% in cost. With a minor adjustment in Exact Scheduling using a globally shared variable $P(t)$, Exact Scheduling PC (partially-centralized) in (31) can reduce the cost for about 10%. This cost reduction leads to less than 10% of difference in cost between Exact Scheduling PC and the fully-centralized Online Optimization (MPC), which requires much more computational and communication resources (Figure 4a). For synthetic instances, 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) only degrades from fully-centralized Online Optimization (MPC) by about 10% on average, and Exact Scheduling (fully-distributed) only degrades from Exact Scheduling PC by another 10% on average. This relation holds beyond the specific arrival distribution considered in Figure 4b-4c (see Figure 10 in the Appendix for the performance comparison in varying arrival distributions).

The relative performance/cost of Exact Scheduling and others depends on the characteristics of the charging instance. To further investigate this dependency, we grouped instances according to its empirical competitive-ratio of performing Exact Scheduling and computed the average arrival rate and demand and sojourn time ratio for each group. In general, empirical competitive-ratio (the comparative performance) improves as the number of arrivals decrease (Figure 5a) and also as average demand to sojourn time ratio increases in size (Figure 5b). These points can also be seen in Figure 9, which compares the service rates for the instance in which the Exact Scheduling performed equally well with Offline Optimal (Figure 9a) and those for the instance in which Exact Scheduling performed much worse (Figure 9b). Intuitively, sparser arrivals would require less coordination between scheduling different jobs, which in turn reduces the advantages of being able to use centralized information. Moreover, when the deadline is tight, the service requirement does not allow much flexibility in varying the service rate over time, and the offline (centralized) algorithm may not be able to use the future arrival information to its full advantage.

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 a few distributed algorithms (Generalized Exact Scheduling, Immediate Scheduling, Equal Service) and centralized algorithms (Earliest Deadline First, Least Laxity First, Fair Sharing) is compared in Figure 6.[13] 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 plateau out at relatively small δ, which results in a lower cost compared with other centralized and distributed algorithms (Figure 6a). 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 6b). This dynamic and optimal adjustment is obtained as the solution of the optimization problem (7), which balances the service capacity variance and unmet demand. Thus, its design process is systematic and does not require tedious manual adjustments.

[13] In Figure 6-7, we use "normalized cost" instead of empirical competitive-ratio. This is because, unlike the case of empirical competitive-ratio, the cost under soft service requirement is compared here with that of offline optimal for hard service requirement.

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 7.[13] Similar to the setting of soft demand, Generalized Exact Scheduling achieves a lower cost than other distributed algorithms (Figure 7a). It also has a comparable performance with the centralized algorithms when the unit penalty for unsatisfied deadline ϵ is large. Such performance is achieved by drastically reducing the total amount of unsatisfied deadline as ϵ increases to avoid the high penalty associated with deadline extension (Figure 7b). This adjustment is obtained as the solution of the optimization problem (8), which systematically balances the service capacity variance and deadline extension.

Generalized Exact Scheduling is the outcome of systematically optimizing service capacity to find the right balance between service capacity variance and the penalties for unsatisfied demands or deadlines. Its excellent performance compared to other distributed algorithms is not surprising because those algorithms are not optimized for dynamic service capacity nor designed to systematically trade-off service capacity variance and unmet demands or deadlines. Moreover, the results in the testbed also suggest that Generalized Exact Scheduling can perform better than other distributed algorithms (such as in the charging of electric vehicles) beyond the case of Poisson arrivals under which Exact Scheduling is optimal.

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, 43]. 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 requirement for all jobs are known, whereas it reduces to Equal Service when the service requirement for none of the jobs are known. With a slight abuse of notation, we denote this extension for case of potentially unknown service requirement as Generalized Exact Scheduling as well.

Now we look into how much system performance may degrade in $Var(P)$ 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

$$\begin{align}
\mathrm{Var}(P) =& \mathbb{E}\left[\int_0^\tau v(\sigma,\tau,y)^2 dy\right] \notag\\
=& p(s=1)\mathbb{E}\left[\frac{\sigma^2}{\tau}\Big| s=1\right] + p(s=0)\mathbb{E}\left[\int_{-\infty}^\tau v(\sigma,\tau,y)^2 d\Big| s=0\right] \tag{33}\\
=& p(s=1)\mathbb{E}\left[\frac{\sigma^2}{\tau}\Big| s=1\right] + p(s=0)\mathbb{E}\left[\frac{\sigma^2}{\tau}\Big| s=0\right] \notag\\
&+ p(s=0)\mathbb{E}\left[\int_{-\infty}^\tau v(\sigma,\tau,y)^2 d\Big| s=0\right] - p(s=0)\mathbb{E}\left[\frac{\sigma^2}{\tau}\Big| s=0\right] \tag{34}\\
=& \mathbb{E}\left[\frac{\sigma^2}{\tau}\right] + p(s=0)\left\{\mathbb{E}\left[\int_{-\infty}^\tau v(\sigma,\tau,y)^2 d\Big| s=0\right] - \mathbb{E}\left[\frac{\sigma^2}{\tau}\Big| s=0\right]\right\} \notag\\
=& q_{ES} + p(s=0)\left\{\mathbb{E}\left[\int_{-\infty}^\tau v(\sigma,\tau,y)^2 dy - (\sigma^2/\tau)\Big| s=0\right]\right\} \tag{35}
\end{align}$$

where q_{ES} is the optimal cost of Exact Scheduling. Equality (33) and (34) use the Law of Total Expectation. From (35), the performance degradation due to unknown demands and deadlines is computed to be[14]

$$\begin{cases} p(s=0)\mathbb{E}\left[{c'_{ES}}^2 \min\{\tau, \sigma/c'_{ES}\} - (\sigma^2/\tau)\right] & \text{in case of soft demand} \\ p(s=0)\mathbb{E}\left[c''_{ES}\sigma - (\sigma^2/\tau)\right] & \text{in case of soft deadline} \end{cases} \tag{36}$$

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 by realizing that the value in (36) is upper-bounded by that of Equal Service for strict demand and deadlines requirement. So the performance difference between Exact Scheduling and Equal Service in Figure 4 can be used to estimate this value.

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 policies 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).

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

$$\text{Var}(X) \leq D \tag{37}$$

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

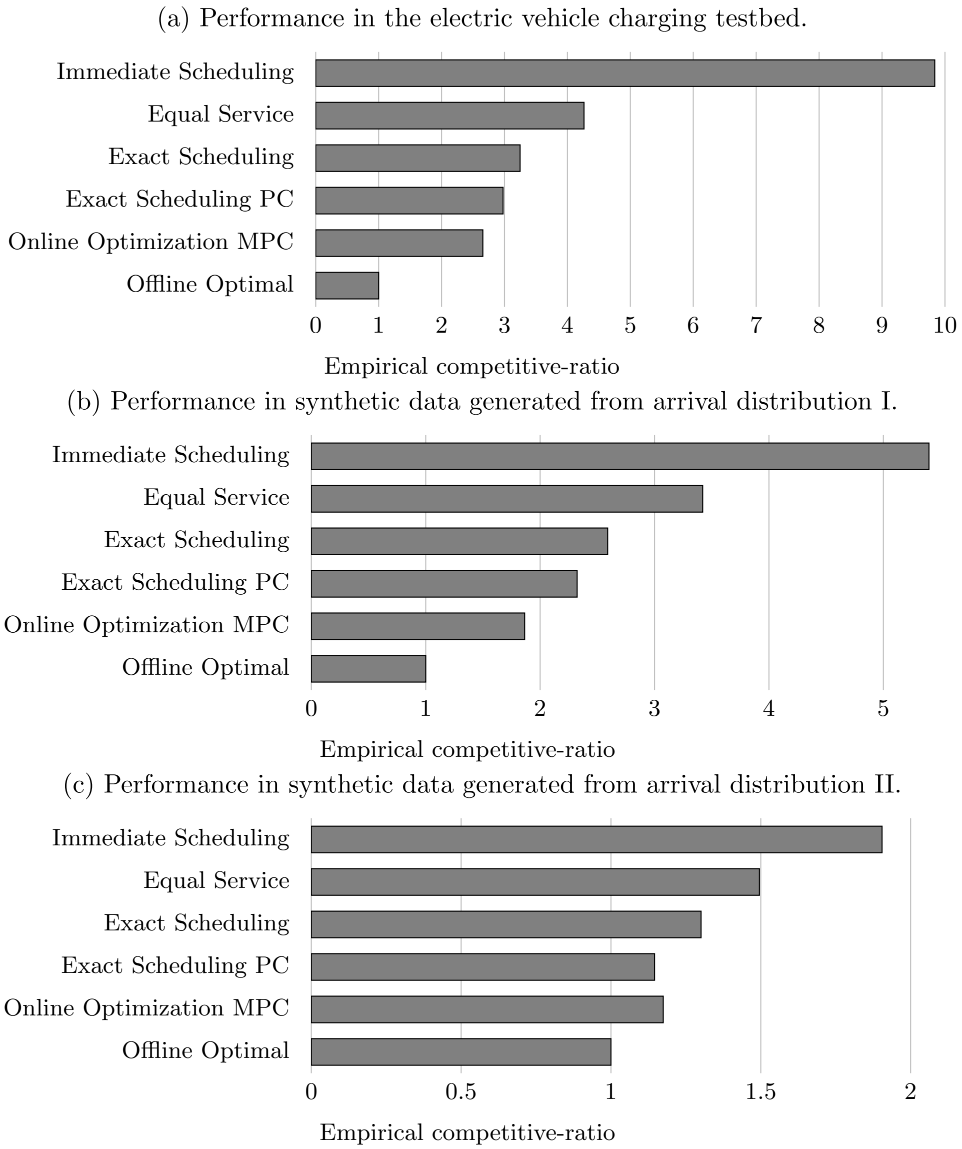
Lemma 3. *Under any centralized policy of the form* (29)*, the stationary variance of $P^\dagger(t)$ is lower-bounded by*

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

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 condition holds:*

$$\text{Var}(P) \leq \frac{4\mathbb{E}\left[\sigma^2/\tau\right]\left(\mathbb{E}[\tau\sigma^2] + \Lambda\mathbb{E}\left[\tau\sigma\right]^2\right)}{\mathbb{E}[\sigma^2]^2}\text{Var}(P^\dagger). \tag{38}$$

[14] 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.



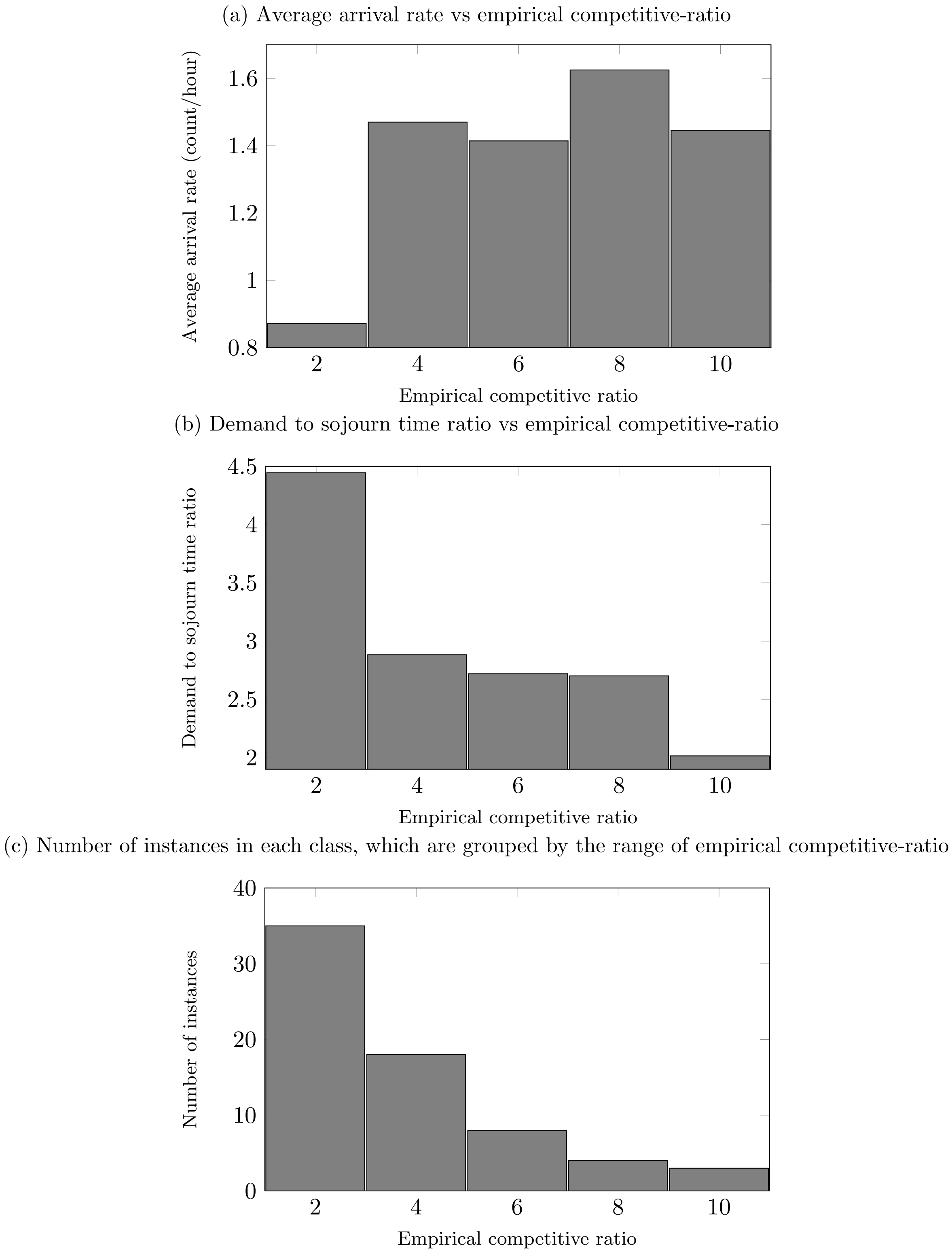
(a) Performance in the electric vehicle charging testbed.

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

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

Figure 4 **Performance comparison of algorithms under strict 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 10 in the Appendix I.**

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 (38) reduces to $\mathrm{Var}(P) \leq 4(1 + \Lambda\tau)\mathrm{Var}(P^{\dagger})$. 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 5a). As $\Lambda\tau \to 0$, the bound suggests that



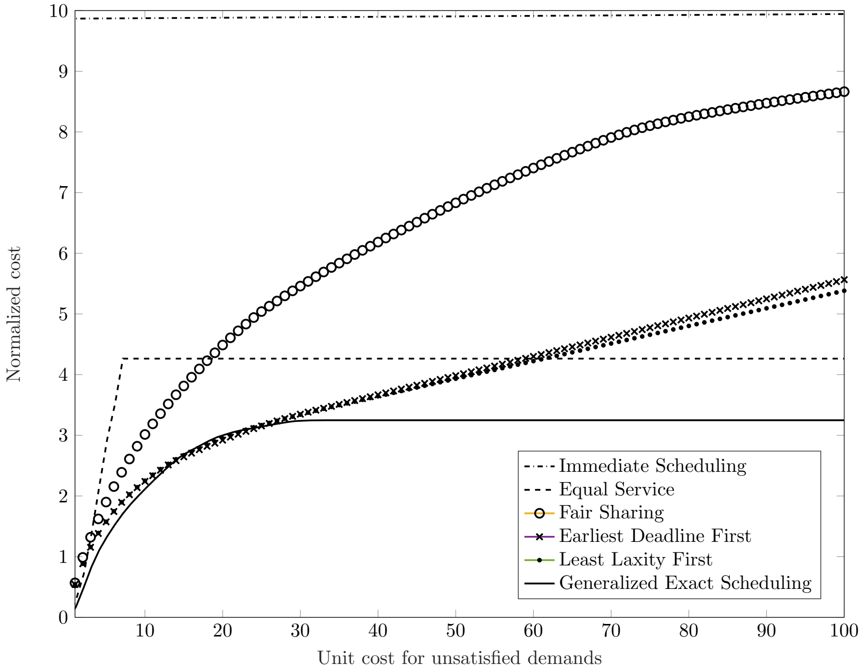
(a) Average arrival rate vs empirical competitive-ratio

(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 5 **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.

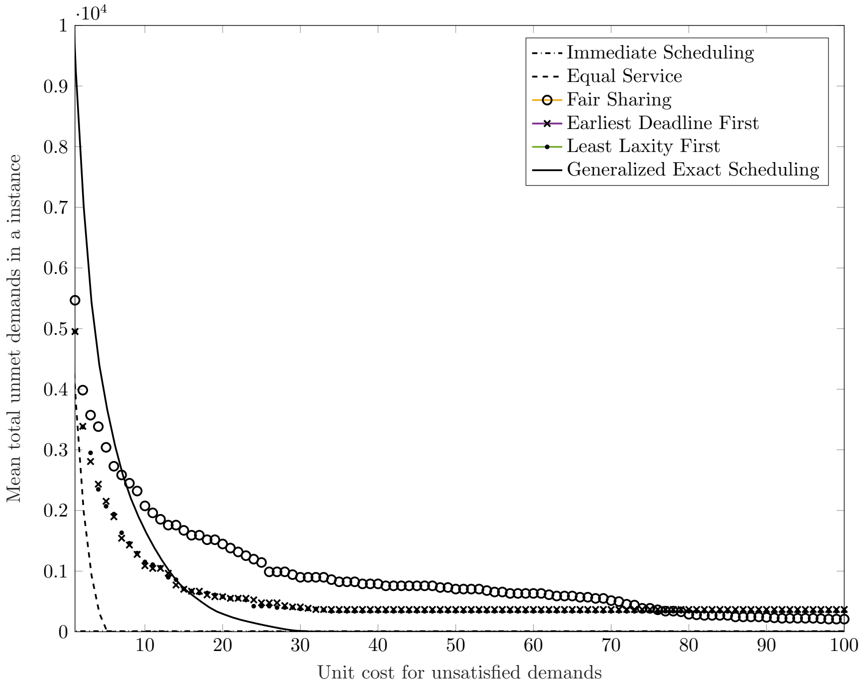
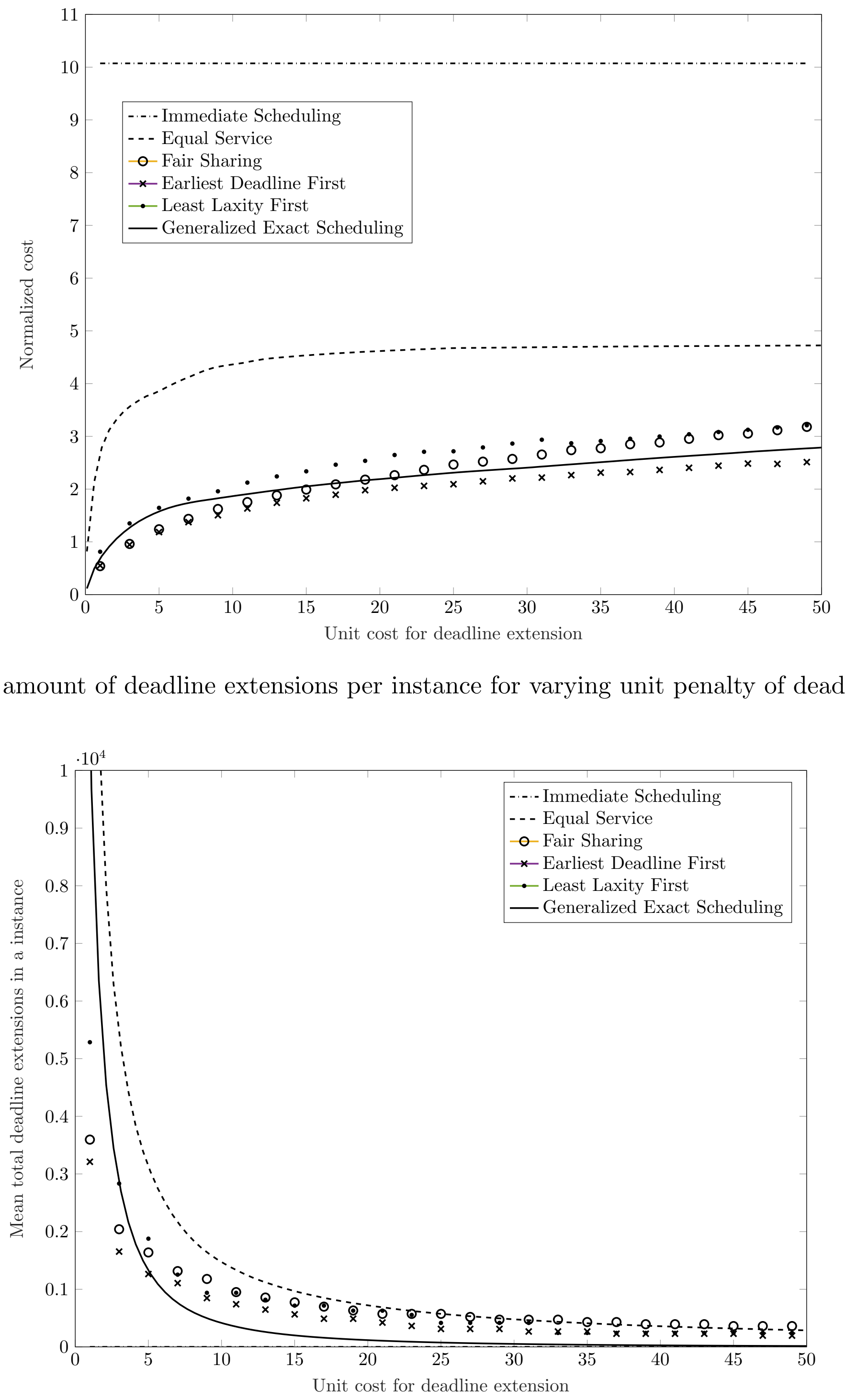


Figure 6 **Generalized Exact Scheduling compared to existing algorithms in the case of soft demand constraints. For varying unit penalty δ, the empirical costs in all instances are shown. The top plot (a) compares the average empirical costs of all instances for varying values unit penalty δ. The cost is normalized by the cost of Offline Optimal (for strict service requirements).[13] The bottom plot (b) shows the average amount of unmet demands in one instance for varying δ. The parameters $c'_{\mathsf{ES}}, p_{\mathsf{EDF}}, p_{\mathsf{LLF}}$, and p_{FS} used in Equal Service, Earliest Deadline First, Least Laxity First, and Fair Sharing are set to be the optimal offline values that minimize the average empirical costs. Note that such optimal offline values require the information of all future instances to be computed, so the estimated performance of these algorithms is optimistic.**


(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 7 **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 requirement).[13] The bottom plot (b) shows the average amount of deadline extension in one instance for varying ϵ. The parameters $p_{\mathsf{EDF}}, p_{\mathsf{LLF}}$, and p_{FS} used in Earliest Deadline First, Least Laxity First, and Fair Sharing are set to be the optimal offline values that minimize the average empirical costs. Note that such optimal offline values require the information of all future instances to be computed, so the estimated performance of these algorithms is optimistic.**

the cost of Exact Scheduling remains within approximately 4 times of the cost of optimal centralized algorithm. Recall from Figure 4a 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 (38) may be due to the fact that the proof of Corollary 2 uses a loose bound for (37) (see (124)–(125) in the Appendix G). Alternatively, a tighter bound can be obtained as

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

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 (37). We consider the following problem:

$$\mathcal{Q}_{\text{on}} = \underset{c:(29)(37)}{\text{minimize}} \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathrm{Var}(P(t))dt,$$

where the optimization is taken over all centralized policies of the form (29) satisfying (37). The Lagrangian of $\mathcal{Q}_{\text{on}}$ is

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

where $\gamma \geq 0$ is the Lagrangian multiplier associated with the constraint (37). Observe that

$$\inf_{c:(29)} L(c;\gamma) \leq \mathcal{Q}_{\text{on}} \leq \mathrm{Var}(P), \tag{39}$$

where $\mathrm{Var}(P)$ is the stationary service capacity variance of any policy. Then, we can derive a lower bound of $\mathrm{Var}(P)$ 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*

$$P(t) = \sqrt{\gamma}(X(t) - \bar{X}) + \bar{P} \tag{40}$$

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

$$\inf_{c:(29)} L(c;\gamma) = \sqrt{\gamma}\Lambda\mathbb{E}[\sigma^2] - \gamma D. \tag{41}$$

Lemma 4 is proven in Appendix F. From (40), the optimal solution of $\inf_{c:(29)} L(c;\gamma)$ satisfies

$$\mathrm{Var}(P) + \gamma\mathrm{Var}(X) = 2\gamma\mathrm{Var}(X). \tag{42}$$

Combining (41) and (42) leads to

$$\mathrm{Var}(X) = \frac{1}{2\sqrt{\gamma}}\Lambda\mathbb{E}[\sigma^2]. \tag{43}$$

Since $X(t)$ also satisfies the constraint (37), the Lagrangian multiplier γ is lower-bounded by

$$\frac{1}{2D}\Lambda\mathbb{E}[\sigma^2] \leq \sqrt{\gamma}. \tag{44}$$

Therefore, we obtain

$$\begin{aligned} \mathrm{Var}(P) &\geq \inf_{c:(29)} L(c;\gamma) \qquad (45)\\ &\geq \frac{\sqrt{\gamma}}{2}\Lambda\mathbb{E}[\sigma^2] \qquad (46)\\ &\geq \frac{1}{4D}\Lambda^2\mathbb{E}[\sigma^2]^2. \qquad (47) \end{aligned}$$

where (45) is due to (39); (46) is due to (41) and (43); and (47) is due to (44).

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

Building upon the results of stationary job arrivals, we consider a more general setting of 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 $f_a(\sigma,\tau)g_a(\delta)h_a(\epsilon)$ (see Section 2). We consider the following three types of policies:

$$r_k(t) = u(a_k, x_k(t), y_k(t)) \geq 0 \quad k \in \mathcal{V} \tag{48}$$

$$r_k(t) = \bar{u}(a_k, x_k(t), y_k(t), \delta_k, \epsilon_k) \geq 0 \quad k \in \mathcal{V} \tag{49}$$

$$r_k(t) = v(a_k, \sigma_k, \tau_k, y_k(t)) \geq 0 \quad k \in \mathcal{V}, \tag{50}$$

In these forms, the scheduling policies to change over time, which allows us account for the changing arrival rate over time. 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.

We seek to design policies that balance three important performance criteria: the quality of service, the service capacity variance associated with the predictability, and its mean square associated with the stability. In the most basic settings involving the first two criteria, we consider the optimization problem

$$\underset{u:(1)(2)(3)(48)}{\text{minimize}} \quad \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathrm{Var}(P(t))dt \tag{51}$$

for the case of strict service requirement and the optimization problem

$$\underset{\bar{u}:(1)(49)}{\text{minimize}} \quad \lim_{T\to\infty} \frac{1}{T}\int_0^T \Big(\mathrm{Var}(P(t)) + \mathbb{E}[U(t)] + \mathbb{E}[W(t)]\Big)dt \tag{52}$$

for the case of soft service requirement. In a more advanced settings involving all three criteria, we consider

$$\underset{v:(1)(2)(3)(50)}{\text{minimize}} \quad \lim_{T\to\infty} \frac{1}{T}\int_0^T \alpha\mathbb{E}[P(t)]^2 + \beta\mathrm{Var}(P(t))dt \tag{53}$$

for the case of strict service requirement. This case also includes

$$\underset{v:(1)(2)(3)(50)}{\text{minimize}} \quad \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathbb{E}[P(t)]^2 dt \tag{54}$$

as an important special case of maximizing stability. More optimization problems can be formulated by combining the terms in (51)–(54). Although such optimization problems are beyond the scope of this paper, our techniques can be used to analyze such problems as well.

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 requirement and soft service requirement. 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 $\mathrm{Var}(P(t))$. In fact, this intuition holds beyond stationary arrivals, and so does for the optimal algorithm for non-stationary arrival distribution.

Theorem 5. *The optimal solution of* (51) *is Exact Scheduling, defined by*

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

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* (52) *is*

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

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 3. 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).

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

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 (48) to (50):

$$\underset{v:(1)(2)(3)(50)}{\text{minimize}} \int_0^T \mathrm{Var}(P(t))dt, \tag{55}$$

where the time horizon T is assumed to be finite. Compared with the stationary setting, obtaining a closed-form solution of (55) requires additional treatment to account for the non-stationarity in arrival distribution.

Lemma 5. *Exact Scheduling* $v(a,\sigma,\tau,y) = (\sigma/\tau)1\{y > 0\}$ *is the optimal solution of* (55).

The proof of Lemma 5 uses the following lemma.

Lemma 6. *The mean and variance of $P(t)$ under the policy* (50) *is given by*

$$\mathbb{E}[P(t)] = \int_{(\sigma,\tau)\in S} \int_0^\tau v(t+y-\tau,\sigma,\tau,y)\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau \tag{56}$$

$$\mathrm{Var}(P(t)) = \int_{(\sigma,\tau)\in S} \int_0^\tau v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau, \tag{57}$$

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

Proof (Lemma 5). From Lemma 6, the objective function of (55) satisfies

$$\begin{aligned}\int_0^T \mathrm{Var}(P(t))dt &= \int_{t=0}^T \int_{(\sigma,\tau)\in S} \int_{y=0}^\tau v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau dt \\ &= \int_{(\sigma,\tau)\in S} \left\{ \int_{y=0}^\tau \int_{t=0}^T v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dtdy \right\} d\sigma d\tau.\end{aligned}$$

Moreover, the constraints of (55) can be rewritten into

$$\int_{y=0}^\tau v(a,\sigma,\tau,y)dy = \sigma \qquad a\in\mathcal{T}, (\sigma,\tau)\in S \tag{58}$$

$$0 \le v(a,\sigma,\tau,y) \le 1 \qquad a\in\mathcal{T}, (\sigma,\tau)\in S, y\in[0,\tau] \tag{59}$$

For any $(\sigma,\tau)\in S$, the optimal solution of (55) is attained at the minimum of the following optimization problem:

$$\underset{v:(58)(59)}{\text{minimize}} \int_{y=0}^\tau \int_{t=0}^T v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dtdy \tag{60}$$

From integration by substitution, the objection function of (60) satisfies

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

where the last equality is due to the assumption that $\Lambda(a,\sigma,\tau)=0$ if $a\notin[0,T-\tau]$. The Lagrangian of (60) is

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

where $\mu_{\sigma,\tau}(a)$ is the Lagrange multiplier associated with constraint (58); $\underline{\nu}_{\sigma,\tau}(a,y)\ge 0$ is the Lagrange multiplier associated with the constraint $v(a,\sigma,\tau,y)\ge 0$, and $\underline{\nu}_{\sigma,\tau}(a,y)$ is the Lagrange multiplier associated with the constraint $\bar{v}(a,\sigma,\tau,y)\le 1$. A necessary condition for v^* to be the optimal scheduling policy is that $L(v;\mu,\nu)$ is stationary at $v=v^*$. After some tedious manipulation, the stationary condition can be computed as follows:

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

We observe that $\underline{\nu}_{\sigma,\tau}(a,y)=0$ when $v^*(a,\sigma,\tau,y)>0$. Combining this condition with (58) and (59) leads to

$$\frac{\mu_{\sigma,\tau}(a)-\bar{\nu}_{\sigma,\tau}(a,y)}{\Lambda(a,\sigma,\tau)}>0.$$

We first suppose $v^*(a,\sigma,\tau,y)=0$ at some $y\in[0,\tau)$. Then for that y, we have

$$\begin{aligned} v^*(a,\sigma,\tau,y)=0&=\frac{\mu_{\sigma,\tau}(a)-\bar{\nu}_{\sigma,\tau}(a,y)}{\Lambda(a,\sigma,\tau)}+\frac{\underline{\nu}_{\sigma,\tau}(a,y)}{\Lambda(a,\sigma,\tau)}\\ &>\frac{\mu_{\sigma,\tau}(a)-\bar{\nu}_{\sigma,\tau}(a,y)}{\Lambda(a,\sigma,\tau)}. \end{aligned}$$

The last inequality cannot hold because the left hand size equals zero while the right hand side is strictly positive. So there is a contradiction, Therefore, $v^*(a,\sigma,\tau,y)$ must take non-zero values at all $y\in[0,\tau)$. We then suppose that $v^*(a,\sigma,\tau,y_1)<v^*(a,\sigma,\tau,y_2)=1$ for some $y_1,y_2\in[0,\tau)$. Then

$$v^*(a,\sigma,\tau,y_1)=\frac{\mu_{\sigma,\tau}(a)}{\Lambda(a,\sigma,\tau)}>\frac{\mu_{\sigma,\tau}(a)-\bar{\nu}_{\sigma,\tau}(a,y)}{\Lambda(a,\sigma,\tau)}=v^*(a,\sigma,\tau,y_2).$$

This is also a contradiction, so $v^*(a,\sigma,\tau,y_1)$ takes a constant value at all $y\in[0,\tau)$. Therefore, the optimal solution of (55) is $v(a,\sigma,\tau,y)=(\sigma/\tau)1\{y>0\}$, which is Exact Scheduling. Q.E.D.

It shall be noted that the optimal value of Exact Scheduling can be represented by control policy of the form (48), and the optimal value of (51) is lower bounded by that of (55). Therefore, Exact Scheduling is also optimal for (51), yielding Theorem 5.

The optimization problem (9) can also be solved in a similar manner. The optimal solution of (51) is also the point-wise 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

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

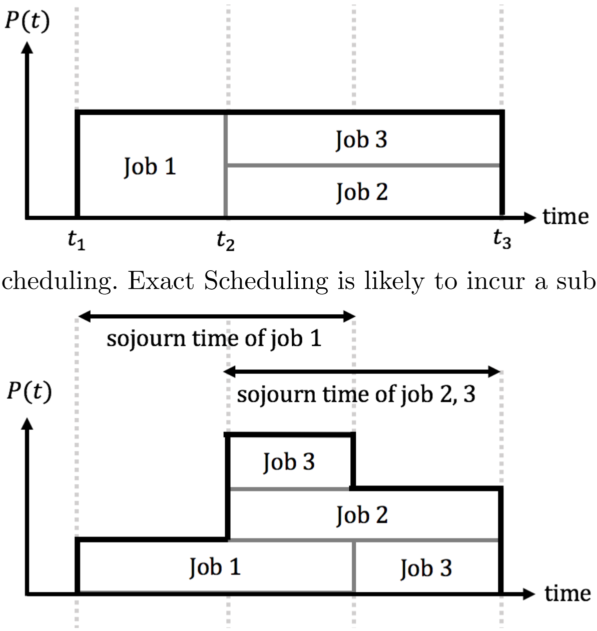
and converting v^* to take the form (50).

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 8. In this instance, the arrival rate increases over time, and Exact Scheduling is likely to incur a substantial cost at a later time (Figure 8a). Meanwhile, an ideal algorithm should account for the increment in future arrivals by serving previous jobs more aggressively than Exact Scheduling (Figure 8).

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



(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 8 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.

Corollary 4. *The optimal solution of* (54) *has the following properties: for each job profiles* $(a,\sigma,\tau)\in(\mathcal{T},S)$*,*

(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 states that a job receives 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 service rate is non-zero because fluctuations of $\mathbb{E}[P(h)]$ during these intervals compromise stability. Condition (ii) picks the period of lowest expected service capacity to serve. In the example of Figure 8, 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 (Figure 8b for an intuitive illustration).

The above property is commonly observed in a well-known class of algorithm: 'Valley Filling' or 'Water Filling'. This class of algorithms was proposed for many budget allocation problems such as CPU scheduling [14], temperature and energy control [6], electric vehicle charging in deterministic settings [43, 56], Parallel Gaussian Channels [57, Chapter 9], optimal packet scheduling [58]. Furthermore, the optimal policy in Corollary 4 also has an interesting similarity to the YDS algorithm [6, 14]. Specifically, the YDS algorithm is the solution of

$$\underset{r\geq 0:(2)(3)}{\text{minimize}}\quad \frac{1}{T}\sum_{t=0}^{T}P(t)^{\alpha} \tag{61}$$

where $\alpha>1$ is some constant. The optimal solution of (61) satisfies the following conditions: for any job $k\in\mathcal{V}$,

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

(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, condition (i)–(ii) in Corollary 4 become condition (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.

Algorithm 1 finds 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 arrive time after t_1 and a deadline before t_2. When $(a,\sigma,\tau) \in \mathcal{V}(t_1,t_2)$, we say that jobs with (a,σ,τ) present in the interval $[t_1,t_2]$. Let $w(t_1,t_2)$ denote the expected cumulative demand of jobs present in the interval $[t_1,t_2]$, *i.e.*

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

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

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

The algorithm 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. At each step, it finds a maximum intensity interval

$$[t_1,t_2] = \arg\max_{[t_1,t_2]} I(t_1,t_2) \tag{62}$$

is computed (line 3). Jobs present in the maximum intensity interval are served subject to

$$\mathbb{E}[P(h)] = \frac{I(t_1,t_2)}{t_2 - t_1} \quad h \in [t_1,t_2) \tag{63}$$

$$v^*(a,\sigma,\tau,y) \leq \bar{v}(a,\sigma,\tau,y). \tag{64}$$

in ascending order of their deadlines (line 4). 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 of jobs not present in $\mathcal{V}(t_1,t_2)$ are set to be zero during $[t_1,t_2]$ (line 6). Because jobs in $\mathcal{V}(t_1,t_2)$ are already scheduled, before the next iteration, they are removed from the arrival statistics $\Lambda(a,\sigma,\tau)$ as if the arrival probability of jobs present in $\mathcal{V}(t_1,t_2)$ is zero (line 7). 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.

We can observe that Algorithm 1 also minimizes $\max_{t\in\mathcal{T}} \mathbb{E}[P(t)]$. This property is derived from the fact that the value of $\max_{t\in\mathcal{T}} \mathbb{E}[P(t)]$ cannot be smaller than what is required to schedule jobs present in $\mathcal{V}(t_1,t_2)$ in the first iteration because the interval $[t_1,t_2]$ found by the first iteration is the most intensive interval of an instance. Moreover, Corollary 4 and the optimal algorithm can be generalized to account for service capacity variance. We discuss this generalization in the next section.

5.4. Balancing stability and predictability

In the previous two sections, we show the optimal policies that maximizes 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.

Algorithm 1: Computing the optimal distributed policy that maximizes stability

```
input  : $\Lambda(a,\sigma,\tau)$
output: $v^*(a,\sigma,\tau,y)$
initialize $\bar{v}(a,\sigma,\tau,y) \leftarrow \infty$;
while $\Lambda(a,\sigma,\tau) > 0$ for some $(a,\sigma,\tau)$ do
    identify the maximum intensity interval $[t_1,t_2]$ by solving (62) ;
    compute $v^*(a,\sigma,\tau,y)$ for job profiles in $\mathcal{V}(t_1,t_2)$ s.t. (63) and (64) ;
    for $(a,\sigma,\tau) \notin \mathcal{V}(t_1,t_2)$ do
        set $\bar{v}(a,\sigma,\tau,y) \leftarrow 0$ for any $a+\tau-y \in [t_1,t_2]$ ;
        set $\Lambda(a,\sigma,\tau) \leftarrow 0$;
```

Recall that, with regard to maximizing predictability, it is favorable to have a fixed service rate over time (see Section 5.2). Meanwhile, with regard 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* (53) *has the following properties: for each job profiles* $(a,\sigma,\tau) \in (\mathcal{T},S)$*,*

(i) $\alpha\mathbb{E}[P(h)] + \beta v(a,\sigma,\tau,a+\tau-h)$ *takes a constant value for any time* h *at which* $v(a,\sigma,\tau,a+\tau-h) > 0$*,*

(ii) $\alpha\mathbb{E}[P(h')] \geq \alpha\mathbb{E}[P(h)] + \beta v(a,\sigma,\tau,a+\tau-h)$ *for any time* $h' \in [a,a+\tau]$ *at which* $v(a,\sigma,\tau,a+\tau-h') = 0$*.*

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 $y \in [0,\tau]$. 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 (Theorem 6) From Lemma 2, the objective function of (54) is equivalent to

$$\begin{aligned}
&\int_0^T \alpha\mathbb{E}[P(t)]^2 + \beta\mathrm{Var}(P(t))dt \\
&= \alpha\int_0^T \left\{\iint_{(\sigma,\tau)\in S}\int_{y=0}^{\tau} v(t+y-\tau,\sigma,\tau,y)\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau\right\}^2 dt \\
&\quad + \beta\int_0^T \int_{(\sigma,\tau)\in S}\int_{y=0}^{\tau} v(t+y-\tau,\sigma,\tau,y)^2\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau dt
\end{aligned}$$

Moreover, the constraints of (53) are equivalent to

$$\int_{y=0}^{\tau} v(a,\sigma,\tau,y)dy = \sigma, \qquad (\sigma,\tau)\in S,\ a\in\mathcal{T} \tag{65}$$

$$v(a,\sigma,\tau,y) \geq 0, \qquad (\sigma,\tau)\in S,\ a\in\mathcal{T},\ y\in[0,\tau]. \tag{66}$$

The Lagrangian associated with problem (54) is

$$L(v;\mu,\nu) = \alpha\int_0^T \left\{\iint_{(\sigma,\tau)\in S}\int_{y=0}^{\tau} v(t+y-\tau,\sigma,\tau,y)\Lambda(t+y-\tau,\sigma,\tau)dyd\sigma d\tau\right\}^2 dt$$

$$
\begin{aligned}
&+\beta\int_{(\sigma,\tau)\in S}\int_{y=0}^{\tau}\int_{a=0}^{T} v(a,\sigma,\tau,y)^2\Lambda(a,\sigma,\tau)dadyd\sigma d\tau\\
&-\int_{a=0}^{T}\int_{(\sigma,\tau)\in S}\mu(\sigma,\tau,a)\int_{y=0}^{\tau} v(y,\sigma,\tau,a)dyd\sigma d\tau da\\
&-\int_{a=0}^{T}\int_{(\sigma,\tau)\in S}\int_{y=0}^{\tau}\nu(a,\sigma,\tau,y)v(y,\sigma,\tau,a)dyd\sigma d\tau da,
\end{aligned}
$$

where $\mu(\sigma,\tau,a)$ is the Lagrange multiplier associated with (65), and $\nu(a,\sigma,\tau,y)\geq 0$ is the Lagrange multiplier associated with (66). We can alternatively consider $L:U\to\mathbb{R}$ as a functional defined on the function space U of policies. Let $U_f\subset U$ be the space of feasible scheduling policies, *i.e.*

$$U_f=\{v : v \text{ satisfies } (65)\&(66)\}.$$

Now we impose an infinitesimal perturbation to v such that

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

Let $G:(\mathcal{T},U)\to\mathbb{R}$ be the following functional:

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

The difference in Lagrangian can be written as

$$
\begin{aligned}
&L(v';\mu,\nu)-L(v;\mu,\nu)\\
=&\alpha\int_{0}^{T}2\,G(t;u)\left\{\int_{(\sigma,\tau)\in S}\int_{a=t-\tau}^{t}\epsilon\tilde{v}(a,\sigma,\tau,y)\Lambda(a,\sigma,\tau)dad\sigma d\tau\right\}dt\\
&+\beta\int_{(\sigma,\tau)\in S}\int_{y=0}^{\tau}\int_{a=0}^{T}2\epsilon\tilde{v}(a,\sigma,\tau,y)v(a,\sigma,\tau,y)\Lambda(a,\sigma,\tau)dadyd\sigma d\tau\\
&-\int_{(\sigma,\tau)\in S}\int_{a=0}^{T}\mu(\sigma,\tau,a)\int_{y=0}^{\tau}\epsilon\tilde{v}(a,\sigma,\tau,y)dydad\sigma d\tau\\
&-\int_{y=0}^{\tau}\nu(\sigma,\tau,a,y)\epsilon\tilde{v}(\sigma,\tau,a,y)dydad\sigma d\tau+O(\epsilon^2)\\
=&\alpha\int_{(\sigma,\tau)\in S}\int_{y=0}^{\tau}\int_{a=0}^{T}2\epsilon G(t;u)\tilde{v}(a,\sigma,\tau,y)\Lambda(a,e,\tau)dadyd\sigma d\tau \qquad (67)\\
&+\beta\int_{(\sigma,\tau)\in S}\int_{y=0}^{\tau}\int_{a=0}^{T}2\epsilon\tilde{v}(a,\sigma,\tau,y)v(a,\sigma,\tau,y)\Lambda(a,\sigma,\tau)dadyd\sigma d\tau\\
&-\int_{(\sigma,\tau)\in S}\int_{a=0}^{T}\int_{y=0}^{\tau}\epsilon(\mu(\sigma,\tau,a)\tilde{v}(a,\sigma,\tau,y)+\nu(\sigma,\tau,a,y)\tilde{v}(\sigma,\tau,a,y))dydad\sigma d\tau+O(\epsilon^2)\\
=&\epsilon\int_{(\sigma,\tau)\in S}\int_{y=0}^{\tau}\int_{a=0}^{T}\big\{2(\alpha G(t;u)+\beta v(a,\sigma,\tau,y))\Lambda(a,\sigma,\tau)-\mu(\sigma,\tau,a)-\nu(\sigma,\tau,a,y)\big\}\tilde{v}(a,\sigma,\tau,y)dadyd\sigma d\tau\\
&+O(\epsilon^2), \qquad (68)
\end{aligned}
$$

where (67) is obtained using integration by substitution. For a functional L to be stationary at some $v\in U_f$, the first term should be zero for any $\tilde{v}(a,\sigma,\tau,y)$ satisfying the constraint $v'\in U_f$. From (68), the stationary point of L satisfies

$$
\begin{aligned}
\alpha\mathbb{E}[P(t)]+\beta v(a,\sigma,\tau,a+\tau-h_1)&=\alpha G(t;u)+\beta v(a,\sigma,\tau,a+\tau-h_1) \qquad (69)\\
&=\frac{\mu(\sigma,\tau,a)+\nu(\sigma,\tau,a,y)}{2\Lambda(a,\sigma,\tau)}
\end{aligned}
$$

for any $y \in [0,\tau]$, $(\sigma,\tau) \in S$, $a \in \mathcal{T}$. Since the optimal solution of (54) is a stationary point of L, (69) is the necessary condition for optimality:

(i) For any job profiles (a,σ,τ), if the service rate is strictly positive at h_1, h_2 such that $h_1 \neq h_1$ and $h_1, h_1 \in [a, a+\tau]$, then

$$v(a-h_1+\tau,\sigma,\tau,a) = v(a-h_2+\tau,\sigma,\tau,a) = 0. \tag{70}$$

Combining (69) and (70) leads to

$$\alpha\mathbb{E}[P(h_1)] + \beta v(a,\sigma,\tau,a+\tau-h_1) = \frac{\mu(\sigma,\tau,a)}{2\Lambda(a,\sigma,\tau)} = \alpha\mathbb{E}[P(h_2)] + \beta v(a,\sigma,\tau,a+\tau-h_2).$$

(ii) For any job profiles (a,σ,τ), if its service rate is strictly positive at $h \in [a,a+\tau]$ and zero at $h' \in [a,a+\tau]$, then

$$\alpha\mathbb{E}[P(h)] + \beta v(a,\sigma,\tau,a+\tau-h_1) = \frac{\mu(\sigma,\tau,a)}{2\Lambda(a,\sigma,\tau)} \leq \frac{\mu(\sigma,\tau,a)+\nu(\sigma,\tau,a,y)}{2\Lambda(a,\sigma,\tau)} = \alpha\mathbb{E}[P(h')].$$

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 an extensive literature analyzing existing algorithms, few analytic results characterizing optimal policies were known in such settings.

In this paper, we characterize the optimal policies in many common scenarios, stationary and non-stationary arrivals, strict or soft demands, with or 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.

When the goal is to balance the stability and predictability, more complex policies turn out to be optimal. Such optimal policies include a novel variation of the YDS algorithm, which is the optimal algorithm for a related worst-case framework. This connection and the proof of optimality suggest a new bridge between the stochastic and worst-case scheduling communities, and this connection will be interesting to be 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 adapt optimal control techniques, which can also be extended to derive performance bonds 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-asynmtotic 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 the context of deadline

The design space of partially-centralized schedulers In Section 4.1, we made an initial attempt to explore the middle ground between fully-centralized algorithm 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 this algorithms, the vast design space of partially-centralized algorithms may have great potential in balancing performance versus scalability but is scarcely explored. Therefore, we note this as an important future direction.

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] Leonard Kleinrock. *Queueing systems, volume I: Theory.* Wiley Interscience, 1975.

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

[18] 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.

[19] 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.

[20] Łukasz Kruk, John Lehoczky, Kavita Ramanan, Steven Shreve, et al. Heavy traffic analysis for edf queues with reneging. *The Annals of Applied Probability*, 21(2):484–545, 2011.

[21] Pascal Moyal. On queues with impatience: stability, and the optimality of earliest deadline first. *Queueing Systems*, 75(2-4):211–242, 2013.

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

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

[24] Erica Plambeck, Sunil Kumar, and Michael J. Harrison. A multiclass queue in heavy traffic with throughput time constraints: Asymptotically optimal dynamic controls. *Queueing Systems*, 39(1):23–54, 2001.

[25] Constantinos Maglaras and Jan A. Van Mieghem. Queueing systems with leadtime constraints: A fluid-model approach for admission and sequencing control. *European journal of operational research*, 167(1):179–207, 2005.

[26] Sabri Çelik and Constantinos Maglaras. Dynamic pricing and lead-time quotation for a multiclass make-to-order queue. *Management Science*, 54(6):1132–1146, 2008.

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

[28] 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.

[29] Qian Zhu and Gagan Agrawal. Resource provisioning with budget constraints for adaptive applications in cloud environments. In *Proc. of the 19th ACM International Symposium on High Performance Distributed Computing*, pages 304–307. ACM, 2010.

[30] Muhammad A. Adnan, Ryo Sugihara, and Rajesh K. Gupta. Energy efficient geographical load balancing via dynamic deferral of workload. In *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, pages 188–195, 2012.

[31] Muhammad A. Adnan and Rajesh K. Gupta. Workload shaping to mitigate variability in renewable power use by data centers. In *2014 IEEE 7th International Conference on Cloud Computing*, pages 96–103, 2014.

[32] 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.

[33] 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.

[34] 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.

[35] Tridib Mukherjee, Ayan Banerjee, Georgios Varsamopoulos, Sandeep KS Gupta, and Sanjay Rungta. Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers. *Computer Networks*, 53(17):2888–2904, 2009.

[36] Yiyu Chen, Amitayu Das, Wubi Qin, Anand Sivasubramaniam, Qian Wang, and Natarajan Gautam. Managing server energy and operational costs in hosting centers. In *ACM SIGMETRICS Performance Evaluation Review*, volume 33 (1), pages 303–314. ACM, 2005.

[37] Luis M Vaquero, Luis Rodero-Merino, and Rajkumar Buyya. Dynamically scaling applications in the cloud. *ACM SIGCOMM Computer Communication Review*, 41(1):45–52, 2011.

[38] 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.

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] Giulio Binetti, Ali Davoudi, David Naso, Biagio Turchiano, and Frank L Lewis. Scalable real-time electric vehicles charging with discrete charging rates. *IEEE Transactions on Smart Grid*, 6(5):2211–2220, 2015.

[46] 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.

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

[48] 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.

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

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

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

[52] 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.

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

[54] 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.

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

[56] 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.

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

[58] 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.

[59] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and 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 [51]). *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:

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

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

$$\int_{(\sigma,\tau)\in S} \int_{\mathbb{R}} v(\sigma,\tau,y)^2 dy \Lambda f(\sigma,\tau) d\sigma d\tau \le \Lambda \int_{(\sigma,\tau)\in S} \sigma f(\sigma,\tau) d\sigma d\tau = \mathbb{E}[\sigma]\Lambda < \infty \tag{71}$$

Combining (71) and Theorem 7, we obtain (56) and (57):

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

which yields Lemma 2.

Appendix B: Proof of Proposition 1

We observe that

$$\begin{aligned}\int_{-\infty}^{\infty} \int_0^{\infty} \frac{\partial}{dy} \lambda(x,y) x dx dy &= \int_0^{\infty} x \left\{ \int_{-\infty}^{\infty} \frac{\partial}{dy} \lambda(x,y) dy \right\} dx & (72)\\ &= -\int_0^{\infty} x \lim_{L\to\infty} \lambda(x,L) dx \\ &= -\Lambda \mathbb{E}[\sigma - \hat{\sigma}(\sigma,\tau)] & (73)\end{aligned}$$

where (72) holds because bounded S implies that $\lambda(x,\infty) = 0$. Therefore, the stationary mean of the service capacity satisfies

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

$$
\begin{aligned}
&= \int_{-\infty}^{\infty}\int_{0}^{\infty} \lambda(x,y)u(x,y)dxdy \\
&= -\int_{-\infty}^{\infty}\int_{0}^{\infty} \frac{d}{dx}\left(\lambda(x,y)u(x,y)\right)xdxdy && (74)\\
&= \int_{-\infty}^{\infty}\int_{0}^{\infty} \left(\frac{\partial}{dy}\lambda(x,y)+\Lambda f(x,y)\right)xdxdy && (75)\\
&= -\Lambda\mathbb{E}[\sigma-\hat{\sigma}(\sigma,\tau)]+\Lambda\mathbb{E}[\sigma] && (76)\\
&= \Lambda\mathbb{E}[\hat{\sigma}(\sigma,\tau)]
\end{aligned}
$$

Here, (74) is due to Integration by Parts, (75) is due to (5), (76) is due to (72)–(73).

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

$$\underset{v:(1)(3)(14)}{\text{minimize}} \ \mathrm{Var}(P)+\mathbb{E}[U]. \tag{77}$$

Because the constraint set of (77) contains that of (7), the optimal value of (77) lower-bounds the optimal value of (7). Therefore, to prove Theorem 2, it suffices to solve (77) (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* (77) *is*

$$v(\sigma,\tau,y)=\min\left\{\frac{\delta}{2},\frac{\sigma}{\tau}\right\}1\left\{y>0\right\}, \tag{78}$$

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

$$\hat{\sigma}(\sigma,\tau)=\int_{-\infty}^{\tau} v(\sigma,\tau,y)dy, \tag{79}$$

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

$$0\le\hat{\sigma}(\sigma,\tau)\le\sigma, \qquad \forall(\sigma,\tau)\in S. \tag{80}$$

Consequently, the stationary mean of U satisfies

$$
\begin{aligned}
\mathbb{E}[U] &= \lim_{t\to\infty}\mathbb{E}\left[\sum_{k\in\mathcal{V}:a_k+\tau_k=t}(\sigma_k-\hat{\sigma}(\sigma_k,\tau_k))\right]\\
&= \int_{(\sigma,\tau)\in S}(\sigma-\hat{\sigma}(\sigma,\tau))\Lambda f(\sigma,\tau)d\sigma d\tau && (81)
\end{aligned}
$$

Then, we use (81) to rewrite (77) as follows

$$
\begin{aligned}
&\inf_{v:(1)(3)(14)} \ \mathrm{Var}(P)+\ \mathbb{E}[U]\\
&= \inf_{\hat{\sigma}:(80)}\left[\inf_{v:(1)(3)(14)(79)}\mathrm{Var}(P)+\delta\int_{(\sigma,\tau)\in S}(\sigma-\hat{\sigma}(\sigma,\tau))\Lambda f(\sigma,\tau)d\sigma d\tau\right] && (82)\\
&= \inf_{\hat{\sigma}:(80)}\left[\left\{\inf_{v:(1)(3)(14)(79)}\mathrm{Var}(P)\right\}+\delta\int_{(\sigma,\tau)\in S}(\sigma-\hat{\sigma}(\sigma,\tau))\Lambda f(\sigma,\tau)d\sigma d\tau\right]. && (83)
\end{aligned}
$$

Equality (83) 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 (82) is not a function of v. From Lemma 1, the first term of (83) admits the closed-form expression

$$\inf_{v:(1)(3)(14)}\mathrm{Var}(P)=\int_{(\sigma,s)\in S}\frac{\hat{\sigma}(\sigma,\tau)^2}{\tau}\Lambda f(\sigma,\tau)d\sigma d\tau, \tag{84}$$

which is attained by

$$v(\sigma,\tau,y)=\frac{\hat{\sigma}(\sigma,\tau)}{\tau}. \tag{85}$$

Substitute (84) into (83) yields

$$\inf_{\hat{\sigma}:(80)}\int_{(\sigma,\tau)\in S}\left\{\frac{\hat{\sigma}(\sigma,\tau)^2}{\tau}+\delta(\sigma-\hat{\sigma}(\sigma,\tau))\right\}\Lambda f(\sigma,\tau)d\sigma d\tau, \tag{86}$$

where the optimization variable is now $\hat{\sigma}$ instead of v. To derive a closed-form solution of (77), we can minimize the integrand of (86) point-wisely. By doing so, we observe that, for each $(\sigma,\tau)\in S$, a necessary and sufficient condition for optimality is

$$\hat{\sigma}(\sigma,\tau)=\arg\inf_{\hat{\sigma}:(80)}\frac{\hat{\sigma}(\sigma,\tau)^2}{\tau}+\delta(\sigma-\hat{\sigma}(\sigma,\tau))=\min\left\{\frac{\delta\tau}{2},\sigma\right\}. \tag{87}$$

Combining (85) and (87), we obtain that (78) is the optimal solution of (77). Substitute (78) into (86), 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 (78) can be realized as (21) using a scheduling policy of the form (21). This implies that the optimal solution of problem (77) also lies within the constraint set of problem (7). Because the cost attained by scheduler (78) 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

$$\underset{v:(1)(2)(14)}{\text{minimize}}\ \operatorname{Var}(P)+\mathbb{E}[W]. \tag{88}$$

Because the optimal value of (88) lower-bounds that of (7), to prove Theorem 3, we can solve (88) (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* (88) *is*

$$v(\sigma,\tau,y)=\begin{cases}\frac{\sigma}{\tau}1\{y>0\} & \text{if } \frac{\sigma}{\tau}\le\sqrt{\epsilon}\\ \sqrt{\epsilon}\,1\left\{y>\tau-\frac{\sigma}{\sqrt{\epsilon}}\right\} & \text{otherwise}\end{cases}. \tag{89}$$

and it achieves the optimal value (24).

Proof. With a slight abuse of notation, let

$$\hat{\tau}(\sigma,\tau)=\begin{cases}\tau & \text{if } \ v(\sigma,\tau,y)=0,\ \forall y<0\\ \tau-\min\{\bar{y}:v(\sigma,\tau,y)=0,\forall y\le\bar{y}\} & \text{otherwise}\end{cases} \tag{90}$$

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 (88) can then be written into

$$\begin{aligned}&\inf_{v:(1)(2)(14)}\ \operatorname{Var}(P)+\mathbb{E}[\epsilon W]\\ &=\inf_{\hat{\tau}\ge\tau}\left[\left\{\inf_{v:(1)(2)(14)}\operatorname{Var}(P)\right\}+\epsilon\int_{(\sigma,\tau)\in S}(\hat{\tau}(\sigma,\tau)-\tau)\Lambda f(\sigma,\tau)d\sigma d\tau\right] &(91)\\ &=\inf_{\hat{\tau}\ge\tau}\int_{(\sigma,\tau)\in S}\left\{\frac{\sigma^2}{\hat{\tau}}+\epsilon(\hat{\tau}(\sigma,\tau)-\tau)\right\}\Lambda f(\sigma,\tau)d\sigma d\tau, &(92)\end{aligned}$$

where $\inf_{v:(1)(2)(14)} \mathrm{Var}(P)$ in (91) is attained by

$$v(\sigma,\tau,y)=\frac{\sigma}{\hat{\tau}(\sigma,\tau)}. \tag{93}$$

The optimal choice of deadline extensions $\hat{\tau}^\star(\sigma,\tau)$ is the point-wise maximum of the integrand of (92), *i.e.*

$$\hat{\tau}^\star(\sigma,\tau)=\arg\inf_{\hat{\sigma}:(80)}\frac{\sigma^2}{\hat{\tau}}+\epsilon(\hat{\tau}(\sigma,\tau)-\tau)=\left\{\frac{\sigma}{\sqrt{\epsilon}},\tau\right\}. \tag{94}$$

Combining (93) and (94), we obtain (89) as the closed-form solution of (88). Q.E.D.

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

Proof (Theorem 3) Recall that the optimal value of problem (88) lower-bounds the optimal value of problem (8). Therefore, if there is a policy of the form (4) that produces identical service rates to (89), 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 $\sigma\le\tau\sqrt{\epsilon}$, both (23) and (89) 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 $\sigma>\sqrt{\epsilon}\tau$, (89) 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

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

where the third inequality is due to $\tau\ge\sqrt{\epsilon}$. Thus, the policy (89) 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.

Appendix E: Proof of Theorem 4

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

$$\underset{v:(1)(14)}{\text{minimize}}\ \mathrm{Var}(P)+\mathbb{E}[U]+\mathbb{E}[W]. \tag{95}$$

The solution of problem (95) 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* (95) *is*

$$v(\sigma,\tau,y)=\begin{cases}\dfrac{\sigma}{\tau}1\{y>0\} & \textit{if } \dfrac{\sigma}{\tau}\le\min\left\{\dfrac{\delta}{2},\sqrt{\epsilon}\right\}\\ \dfrac{\delta}{2}1\{y>0\} & \textit{if } \dfrac{\sigma}{\tau}>\dfrac{\delta}{2} \textit{ and } \dfrac{\delta}{2}\le\sqrt{\epsilon}\\ \sqrt{\epsilon}1\left\{y>\tau-\dfrac{\sigma}{\sqrt{\epsilon}}\right\} & \textit{otherwise}\end{cases}. \tag{96}$$

and it achieves the optimal value (26)*.*

Proof. Recall that $\hat{\sigma}(\sigma,\tau)$ in (79) denotes the actual service supply for jobs having a service demand σ and a sojourn time τ, and $\hat{\tau}(\sigma,\tau)$ in (90) denote the actual sojourn time for such jobs. The optimization problem (95) can be written into

$$\begin{aligned}&\inf_{v:(1)(14)}\ \mathrm{Var}(P(t))+\mathbb{E}[\delta U]+\mathbb{E}[\epsilon W]\\ &=\inf_{\substack{\hat{\sigma}(\sigma,\tau)\ge\sigma\\ \hat{\tau}(\sigma,\tau)\ge\tau}}\left[\inf_{v:(1)(14)}\mathrm{Var}(P)+\int_{(\sigma,\tau)\in S}\{\delta(\sigma-\hat{\sigma}(\sigma,\tau))+\epsilon(\hat{\tau}(\sigma,\tau)-\tau)\}\Lambda f(\sigma,\tau)d\sigma d\tau\right]\\ &=\inf_{\substack{\hat{\sigma}(\sigma,\tau)\ge\sigma\\ \hat{\tau}(\sigma,\tau)\ge\tau}}\int_{(\sigma,\tau)\in S}\left[\frac{\hat{\sigma}(\sigma,\tau)^2}{\hat{\tau}(\sigma,\tau)}+\delta(\sigma-\hat{\sigma}(\sigma,\tau))+\epsilon(\hat{\tau}(\sigma,\tau)-\tau)\right]\Lambda f(\sigma,\tau)d\sigma d\tau\\ &=\inf_{\substack{\hat{\sigma}(\sigma,\tau)\ge\sigma\\ \hat{\tau}(\sigma,\tau)\ge\tau}}\int_{(\sigma,\tau)\in S}C(\sigma,\hat{\tau})\Lambda f(\sigma,\tau)d\sigma d\tau,\end{aligned} \tag{97}$$

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

$$C(\sigma,\hat{\tau}) := \frac{\hat{\sigma}(\sigma,\tau)^2}{\hat{\tau}(\sigma,\tau)} + \delta(\sigma - \hat{\sigma}(\sigma,\tau)) + \epsilon(\hat{\tau}(\sigma,\tau) - \tau)$$
$$= \begin{cases} \dfrac{\sigma^2}{\tau} & \text{if } \hat{\tau} = \tau \text{ and } \dfrac{\sigma}{\tau} \le \dfrac{\delta}{2} \\ \delta\left(\sigma - \dfrac{\delta\tau}{4}\right) & \text{if } \hat{\tau} = \tau \text{ and } \dfrac{\sigma}{\tau} > \dfrac{\delta}{2} \\ \dfrac{\sigma^2}{\hat{\tau}} + \epsilon(\hat{\tau} - \tau) & \text{if } \hat{\tau} > \tau \text{ and } \dfrac{\sigma}{\hat{\tau}} \le \dfrac{\delta}{2} \\ \delta\left(\sigma - \dfrac{\delta\hat{\tau}}{4}\right) + \epsilon(\hat{\tau} - \tau) & \text{if } \hat{\tau} > \tau \text{ and } \dfrac{\sigma}{\hat{\tau}} > \dfrac{\delta}{2} \end{cases}.$$

Relation (97) 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 (97).

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

$$\begin{aligned} C(\sigma,\hat{\tau}) - C(\sigma,\tau) &= \frac{\sigma^2}{\hat{\tau}} + \epsilon(\hat{\tau} - \tau) - \frac{\sigma^2}{\tau} \\ &= (\hat{\tau} - \tau)\left(\epsilon - \frac{\sigma^2}{\tau\hat{\tau}}\right) \\ &\ge (\hat{\tau} - \tau)\left\{\epsilon - \left(\frac{\delta\tau}{2}\right)^2 \frac{1}{\tau\hat{\tau}}\right\} && (98) \\ &\ge (\hat{\tau} - \tau)\left\{\epsilon - \frac{\delta^2}{4}\right\} && (99) \\ &\ge 0, && (100) \end{aligned}$$

where (98) is due to $\sigma \le \delta\tau/2$; (99) is due to $\hat{\tau} > \tau$; and (100) is due to $\delta^2/4 \le \epsilon$. When $\sigma \in (\delta\tau/2, \delta\hat{\tau}/2]$, we have

$$\begin{aligned} C(\sigma,\hat{\tau}) - C(\sigma,\tau) &= \frac{\sigma^2}{\hat{\tau}} + \epsilon(\hat{\tau} - \tau) - \delta\left(\sigma - \frac{\delta\tau}{4}\right) \\ &\ge \epsilon(\hat{\tau} - \tau) + \left(\frac{\delta\tau}{2}\right)^2 \frac{1}{\hat{\tau}} - \delta\frac{\delta\hat{\tau}}{2} + \frac{\delta^2\hat{\tau}}{4} && (101) \\ &\ge \epsilon(\hat{\tau} - \tau) + \frac{1}{2}\delta^2(\tau - \hat{\tau}) && (102) \\ &= (\hat{\tau} - \tau)\left\{\epsilon - \frac{\delta^2}{4}\right\} \\ &\ge 0, && (103) \end{aligned}$$

where (101) is due to $\sigma \le \delta\tau/2$; (102) is due to $\hat{\tau} > \tau$; and (103) is due to $\delta^2/4 \le \epsilon$. When $\sigma > \delta\hat{\tau}/2$, we have

$$\begin{aligned} C(\sigma,\hat{\tau}) - C(\sigma,\tau) &= \delta\left(\sigma - \frac{\delta\hat{\tau}}{4}\right) + \epsilon(\hat{\tau} - \tau) - \delta\left(\sigma - \frac{\delta\tau}{4}\right) \\ &= (\hat{\tau} - \tau)\left(\epsilon - \frac{\delta^2}{4}\right) \\ &\ge 0 && (104) \end{aligned}$$

where (104) is due to $\delta^2/4 \le \epsilon$. Since (100), (103), and (104) contradict with 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 (96) as the closed-form solution of (95). Q.E.D.

Theorem 4 is an immediate consequence of Lemma 9. To see it, recall that the optimal value of problem (95) lower-bounds that of problem (9). Moreover, scheduler (25) of the form (4) can produce identical service rates to (96), 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

$$\inf_{c:(29)} L(c;\gamma) = \inf_{c:(29)} \lim_{T\to\infty} \frac{1}{T}\int_0^T \mathrm{Var}(P(t)) + \gamma(\mathrm{Var}(X(t)) - D)dt \tag{105}$$

$$\geq \inf_{c:(29)} \lim_{T\to\infty} \inf_{c:(29)} \frac{1}{T}\int_0^T \mathrm{Var}(P(t)) + \gamma(\mathrm{Var}(X(t)) - D)dt \tag{106}$$

$$= \lim_{T\to\infty} \inf_{c:(29)} \frac{1}{T}\int_0^T \mathrm{Var}(P(t)) + \gamma(\mathrm{Var}(X(t)) - D)dt, \tag{107}$$

where equality (105) holds by the definition of $L(c;\gamma)$, inequality (106) 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 (107) 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

$$X(t_{n+1}) = X(t_n) + A(t_n, h) - hP(t_n), \tag{108}$$

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 $t_1, t_2 \in [t_n, t_{n+1})$,[15] so the service capacity takes the constant value $P(t_n)$ during this interval.

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

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

Observe that (107) satisfies

$$\begin{aligned}
&\lim_{T\to\infty} \inf_{c:(29)} \frac{1}{T}\int_0^T \mathbb{E}[(P(t) - \bar{P})^2 + \gamma((X(t) - \bar{X})^2 - D)]dt \\
&= \lim_{T\to\infty} \inf_{c:(29)} \lim_{h\to 0} \frac{1}{T} L_{h,\lceil T/h\rceil}(c;\gamma)h - \gamma D \\
&= \lim_{T\to\infty} \lim_{h\to 0} \inf_{c:(29)} \frac{1}{T} L_{h,\lceil T/h\rceil}(c;\gamma)h - \gamma D.
\end{aligned} \tag{109}$$

To solve (109), we first consider the cost-to-go $J_n(X(t_n))$ for some $h > 0$ and $N \in \mathbb{Z}_+$, *i.e.*

$$J_n(X(t_n)) := \mathbb{E}\left[\gamma(X(t_N) - \bar{X})^2\right] + \sum_{k=n}^{N-1} \mathbb{E}\left[(P(t_k) - \bar{P})^2 + \gamma(X(t_k) - \bar{X})^2\right]. \tag{110}$$

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

$$J_n(X(t_n)) = \mathbb{E}\left[p_n(X(t_n) - \bar{X})^2\right] + \sum_{k=n}^{N-1} \mathbb{E}[p_{k+1}(A(t_n, h) - \bar{A}_h)^2], \tag{111}$$

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

$$p_k = p_{k+1} - \frac{h^2 p_{k+1}^2}{h^2 p_{k+1} + 1} + \gamma, \qquad p_N = \gamma. \tag{112}$$

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

First, condition (111) holds for $n=N$ by the construction of (110). Second, assume that condition (111) holds for $n+1$. Let $\bar{A}_h$ be the stationary mean of $A(t_n,h)$. Recall from (110) 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

$$\begin{aligned}
&J_n(X(t_n))\\
&=\inf_{P(t_n)}\mathbb{E}[(P(t_n)-\bar{P})^2+\gamma(X(t_n)-\bar{X})^2+J_{n+1}(X(t_{n+1}))] && (113)\\
&=\inf_{P(t_n)}\mathbb{E}[(P(t_n)-\bar{P})^2+\gamma(X(t_n)-\bar{X})^2+J_{n+1}(X(t_n)+A(t_n,h)-hP(t_n)] && (114)\\
&=\inf_{P(t_n)}\mathbb{E}[(P(t_n)-\bar{P})^2+\gamma(X(t_n)-\bar{X})^2+J_{n+1}(X(t_n)+(A(t_n,h)-\bar{A}_h)-h(P(t_n-\bar{P}))] && (115)\\
&=\inf_{P(t_n)}\mathbb{E}[(P(t_n)-\bar{P})^2+\gamma(X(t_n)-\bar{X})^2+p_{n+1}(X(t_n)+(A(t_n,h)-\bar{A}_h)-h(P(t_n)-\bar{P}))^2] && (116)\\
&\qquad+\sum_{k=n+1}^{N-1}\mathbb{E}[p_{k+1}(A(t_k,h)-\bar{A}_h)^2]
\end{aligned}$$

where (114) uses relation (108); (115) relies on $\bar{A}_h=h\bar{P}$ from Brumelle's formula; (116) uses the induction hypothesis that the cost-to-go at $n+1$ satisfies (111). Note that

$$\mathbb{E}[(A(t_n,h)-\bar{A}_h)X(t_n)]=\mathbb{E}[A(t_n,h)-\bar{A}_h]\mathbb{E}[X(t_n)]=0, \tag{117}$$

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 (113) and applying $\mathbb{E}[(A(t_n,h)-\bar{A}_h)X(t_n)]=0$, we can rewrite (117) into

$$\begin{aligned}
J_n(X(t_n))=&(p_{n+1}+\gamma)(X(t_n)-\bar{X})^2+\sum_{k=n}^{N}p_{k+1}\mathbb{E}(A(t_k,h)-\bar{A}_h)^2]\\
&+\inf_{P(t_n)}\{(1+h^2p_{n+1})(P(t_n)-\bar{P})^2-2h\gamma p_{n+1}(X(t_n)-\bar{X})(P(t_n)-\bar{P})]. && (118)
\end{aligned}$$

The minimum value of (118) is attained by

$$P(t_n,h)-\bar{P}_h=\frac{hp_n}{1+h^2p_n}(X(t_n)-\bar{X}), \tag{119}$$

and the optimal cost-to-go becomes (111), where p_n is defined by (112). As $N\to\infty$, p_k converges to a unique positive scalar

$$p:=\lim_{N\to\infty}p_k=\frac{h^2\gamma+h\sqrt{\gamma}\sqrt{h^2\gamma+4}}{2h^2}, \tag{120}$$

which is also a fixed point of (112) [59]. Taking the limit of $N\to\infty$ and $h\to 0$ for (119) and (120), the infimum of (109) is attained when

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

Finally, the infimum value of (109) is computed as

$$\begin{aligned}
&\lim_{T\to\infty}\inf_{c:(29)}\frac{1}{T}\int_0^T\mathrm{Var}(P(t))+\gamma(\mathrm{Var}(X(t))-D)dt\\
&=\lim_{T\to\infty}\lim_{h\to 0}\frac{h}{T}\sum_{k=0}^{N-1}\mathbb{E}[p_{k+1}(A(t_k,h)-\bar{A}_h)^2]-\gamma D && (121)\\
&=\sqrt{\gamma}\Lambda\mathbb{E}[\sigma^2]. && (122)
\end{aligned}$$

where equality (121) is due to (111); and equality (122) is derived from (120).

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.*

$$\begin{aligned} A(t,h) &:= \sum_{\{k\in\mathcal{V}: a_k\in[t,t+h)\}} \sigma_k, \\ P(t,h) &:= \int_t^{t+h} P(\tau)d\tau. \end{aligned}$$

let $D_t = \{k \in \mathcal{V} : a_k + \tau_k \le t\}$ 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

$$\begin{aligned} X(t) &= \sum_{k\in A_t} \sigma_k - \int_{\tau\ge t} P(\tau)d\tau \\ &\le \sum_{k\in A_t} \sigma_k - \sum_{i\in D_t} \sigma_k \\ &\le \sum_{k\in A_t\setminus D_t} \sigma_k \qquad (123) \end{aligned}$$

where $D_t = \{k \in \mathcal{V} : a_k + \tau_k \le t\}$ is the set of jobs that departs by time t. From (123) and $X(t) \ge 0$, the variance of $X(t)$ is upper-bounded by

$$\begin{aligned} \mathrm{Var}(X(t)) &\le \mathbb{E}[X(t)^2] \qquad (124) \\ &\le \mathbb{E}\left[\left(\sum_{k\in A_t\setminus D_t} \sigma_k\right)^2\right] \\ &= \mathrm{Var}\left(\sum_{k\in A_t\setminus D_t} \sigma_k\right) + \mathbb{E}\left[\sum_{k\in A_t\setminus D_t} \sigma_k\right]^2 \\ &= \int_{(\sigma,\tau)\in S} \tau\sigma^2\Lambda f(\sigma,\tau)d\sigma d\tau + \left(\int_{(\sigma,\tau)\in S} \tau\sigma\Lambda f(\sigma,\tau)d\sigma d\tau\right)^2 \\ &= \Lambda\mathbb{E}\left(\tau\sigma^2\right) + \left(\Lambda\mathbb{E}\left[\tau\sigma\right]\right)^2 \qquad (125) \end{aligned}$$

Applying $D = \Lambda\mathbb{E}\left(\tau\sigma^2\right) + \left(\Lambda\mathbb{E}\left[\tau\sigma\right]\right)^2$ to Lemma 3, we obtain (38).

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,

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

we obtain a competitive-ratio like bound for Exact Scheduling (13).

Corollary 5. *Let* $\mathrm{Var}(P)$ *be the stationary variance of* $P(t)$ *that is attained by Exact Scheduling* (13)*. Let* $\mathrm{Var}(P^\dagger)$ *be the minimum stationary variance attainable by any centralized algorithm* (29) *with the same level of* $\mathrm{Var}(X)$ *as Exact Scheduling. Then, the following condition holds:*

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

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

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

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

Corollary 6. *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:*

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

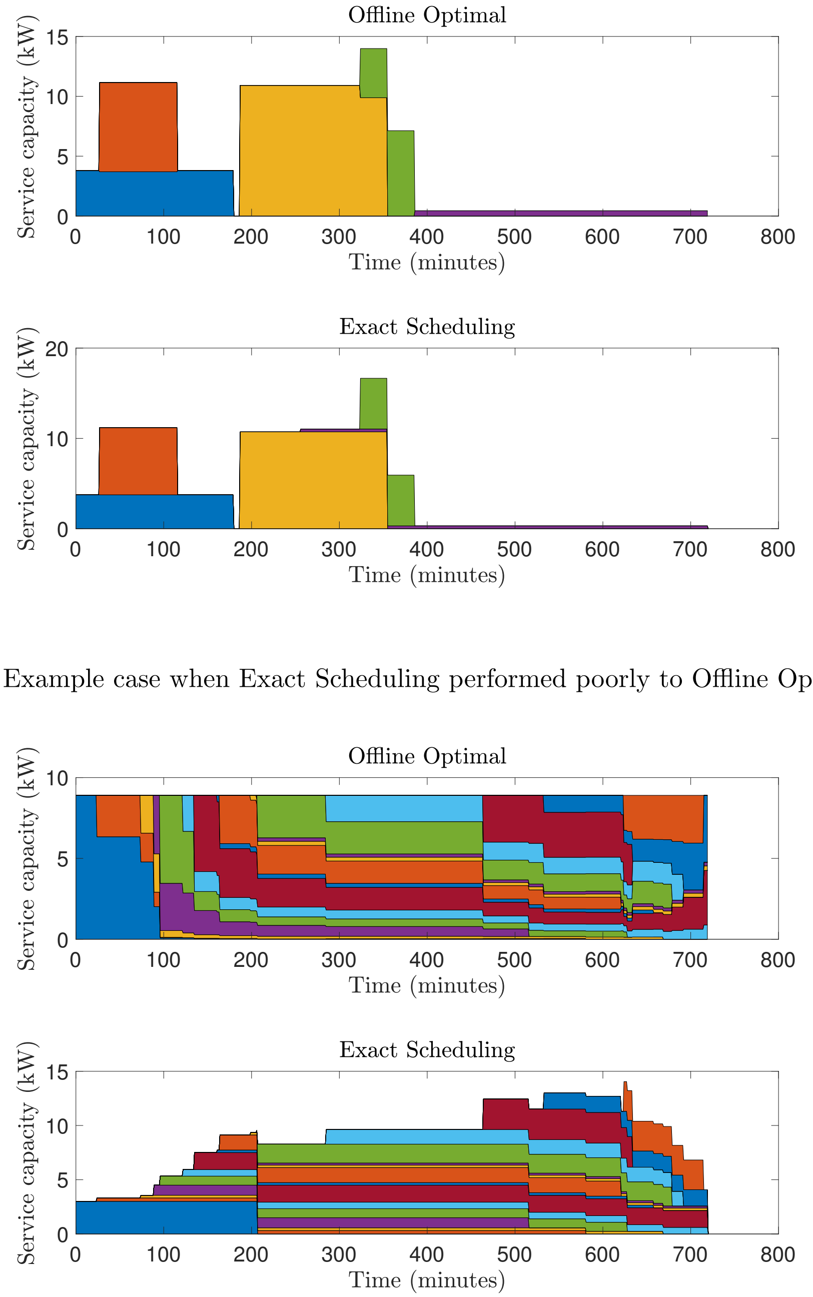
where

$$\begin{aligned}\alpha =& \mathbb{E}\left[\frac{\sigma^2}{\tau}\mathbf{1}\left\{\frac{\sigma}{\tau}\le\min\left\{\frac{\delta}{2},\sqrt{\epsilon}\right\}\right\}+\delta\left(\sqrt{\epsilon}-\frac{\delta\tau}{4}\right)\mathbf{1}\left\{\frac{\sigma}{\tau}>\frac{\delta}{2}\ge\sqrt{\epsilon}\right\}+\left(2\sqrt{\epsilon}\sigma-\epsilon\tau\right)\mathbf{1}\left\{\frac{\sigma}{\tau}>\sqrt{\epsilon}>\frac{\delta}{2}\right\}\right]\\ \beta =& \mathbb{E}\left[\frac{\sigma^2\tau}{3}\mathbf{1}\left\{\frac{\sigma}{\tau}\le\min\left\{\frac{\delta}{2},\sqrt{\epsilon}\right\}\right\}+\left(\frac{\delta^2\tau^3}{12}-\frac{1}{2}\delta\sigma\tau^2+\sigma^2\tau\right)\mathbf{1}\left\{\frac{\sigma}{\tau}>\frac{\delta}{2}\ge\sqrt{\epsilon}\right\}+\left(\frac{\sigma^3}{3\sqrt{\epsilon}}\right)\mathbf{1}\left\{\frac{\sigma}{\tau}>\sqrt{\epsilon}>\frac{\delta}{2}\right\}\right].\end{aligned}$$

Corollary 6 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

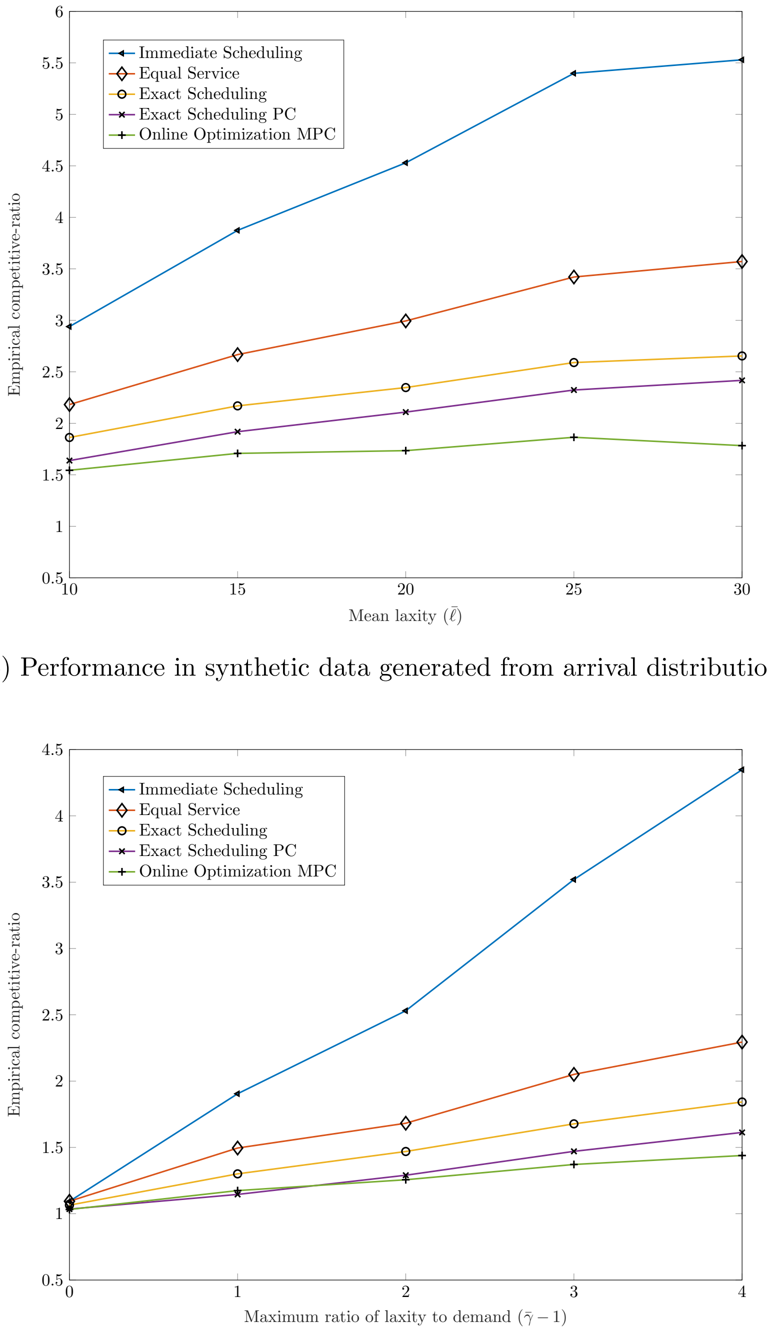
Section 4.1 shows the empirical performance of different algorithms for typical cases. In this section, we provide more detailed experimental data to support the results in Section 4.1. Figure 9 compares how Exact Scheduling and Offline Optimal schedule jobs in two instances: one instance in which Exact Scheduling performed competitively, and another instance in which Exact Scheduling performed poorly. Figure 10 provide a more comprehensive view of Figure 4 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 9 **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 10 **Performance comparison of algorithms under strict 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 Figure 4b. In plot (b), the maximum ratio of laxity to demand refers to $\bar{\gamma} - 1$ in distribution II, and the empirical competitive-ratio for $\bar{\gamma} - 1 = 1$ is shown in Figure Figure 4c.**