Minimal-Variance Distributed Deadline Scheduling in a Stationary Environment

Yorie Nakahira
California Institute of Technology
Pasadena, CA
ynakahir@caltech.edu

Andres Ferragut
Universidad ORT Uruguay
Montevideo, Uruguay
ferragut@ort.edu.uy

Adam Wierman
California Institute of Technology
Pasadena, CA
adamw@caltech.edu

ABSTRACT

Many modern schedulers can dynamically adjust their service capacity to match the incoming workload. At the same time, however, variability in service capacity often incurs operational and infrastructure costs. In this paper, we propose distributed algorithms that minimize service capacity variability when scheduling jobs with deadlines. Specifically, we show that *Exact Scheduling* minimizes service capacity variance subject to strict demand and deadline requirements under stationary Poisson arrivals. We also characterize the optimal distributed policies for more general settings with soft demand requirements, soft deadline requirements, or both. Additionally, we show how close the performance of the optimal distributed policy is to that of the optimal centralized policy by deriving a competitive-ratio-like bound.

KEYWORDS

Deadline scheduling, Service capacity control, Exact Scheduling, Online algorithms

ACM Reference Format:
Yorie Nakahira, Andres Ferragut, and Adam Wierman. 2018. Minimal-Variance Distributed Deadline Scheduling in a Stationary Environment. In *Proceedings of IFIP WG 7.3 Performance 2018 (IFIP Performance 2018).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.475/123_4

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, when using cloud computing services, one can modify the total computing capacity by changing the number of computing instances and their speeds. Power distribution networks can also adapt the energy supply to match the energy demand as it changes over time.

The ability to adapt service capacity dynamically gives rise to challenging new design questions. In particular, how to reduce the *variability* of service capacity is of great importance in such applications since peaks and fluctuations often come with significant costs [8, 23, 30]. This trend is especially true for the examples of cloud computing and power distribution networks mentioned above. Cloud content providers 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. The emerging load from electric vehicle charging stations also leads to similar challenges in power distribution networks. Charging stations require stability in power consumption because fluctuations and 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 because purchasing power in real time is typically more expensive than purchasing in advance.

Thus, in situations where service capacity can be dynamically adjusted, an important design goal is to reduce the costs associated with variability in the service capacity while maintaining a high quality of service. In this work, we study this problem by minimizing the *variance* of the service capacity in systems where jobs arrive with demand and deadline requests. Our focus on service capacity variance is motivated by applications such as cloud computing and power distribution networks, where contracts often explicitly depend on service capacity variability, e.g., if a charging station participates in the regulation market, then costs/payments depend explicitly on the variance of the total capacity [3, 29].

The goal of this work is to design distributed scheduling algorithms that minimize the variance of service capacity subject to service quality constraints, e.g., meeting job deadlines and satisfying job demands. Our focus is on *distributed* scheduling algorithms since implementing centralized algorithms is likely to be prohibitively slow and costly in large-scale service systems today. From cloud computing to power distribution networks, such systems are unlikely to be able to access global information about every job and server in the system when deciding the service rate of each job/server. Therefore, distributed algorithms are a necessity to enable large-scale implementation.

Related work. Although the literature on deadline scheduling is large and varied, optimal algorithms are only known for certain niche cases. Examples of classic scheduling algorithms include Earliest Deadline First [16, 24] and Least Laxity First [16], among others. Beyond these classic algorithms, more modern algorithms simultaneously perform admission control and service rate control in order to exploit the flexibility arising from soft demand or deadline requirements, e.g., [9, 22, 28]. The trade-offs between service quality and costs associated with variability have become a focus only recently, but already many interesting results have appeared, contrasting the performance of classical algorithms, e.g., [7, 12, 13]. These issues have also been studied extensively

IFIP Performance 2018, December 2018, Toulouse, France

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

in the areas of cloud computing, where algorithms have been proposed to control the variability of power usage in data centers using deferrable jobs (see [11, 15, 21, 32] and the references therein), and power distribution systems, where algorithms have been designed to control the variability of energy supply using deferrable loads (see [6, 10, 14, 25, 31] and the references therein).

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 [5, 26, 27] and deterministic worst-case settings [2].

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. Further, we provide a competitive-ratio-like bound that describes the gap between the performance of an optimal distributed policy and the performance of an optimal centralized policy.

Specifically, we identify the optimal distributed algorithms in settings with stationary Poisson job arrivals under strict service requirements (Theorem 3.1), soft demand requirements (Theorem 3.2), soft deadline requirements (Theorem 3.3), and soft demand and deadline requirements (Theorems 3.4). In the most classical setting of strict service requirements, we show that *Exact Scheduling* is the optimal distributed algorithm that minimizes the stationary variance of the service capacity. Exact Scheduling is a classical algorithm that works by finishing job service *exactly* at their deadlines using a constant service rate [8, 13, 20]. In the settings of soft service requirements, we derive the optimal algorithms that minimize a weighted sum of the service capacity variance and the expected penalties for unsatisfied demands and/or deadlines. These algorithms all have closed-form expressions. Moreover, they all use constant service rates and can be considered as generalizations of Exact Scheduling which make use of varying forms of rate and admission control.

Given that our results focus on distributed algorithms, an important question is how these distributed algorithms perform compared with the optimal centralized algorithm, which may provide better performance in theory but requires prohibitively expensive computation to find in practice. To answer this question, we derive a closed-form bound on the performance degradation due to using a distributed algorithm in the setting of strict service constraints (Corollary 4.2). The bound suggests that, when sojourn times are homogeneous (the sojourn time is a deterministic variable), 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 4.1) is novel in its use of optimal control and has the potential for providing competitive-ratio-like bounds for other scheduling policies. We also contrast distributed algorithms with centralized algorithms in the context of one of our motivating examples, electric vehicle charging. Using public data from an Electric Vehicle Charging Testbed [17], we show that the optimal distributed algorithms we propose also achieve comparable performance with existing centralized algorithms in practice.

2 SYSTEM MODEL

The goal of this paper is to characterize online scheduling policies for systems with the ability to dynamically adjust their service capacity which minimize the service capacity variability while satisfying the service requirements (demands and deadlines) of individual jobs. Specifically, we consider a setting in which a service system may scale its capacity in order to serve jobs that randomly arrive with heterogeneous service requirements. We use a continuous time model and use $t \in \mathbb{R}_+$ to denote a point in time. Each job, indexed by $k \in \mathcal{V} = \{1, 2, \ldots\}$, is characterized by a random arrival time a_k, a random service demand σ_k, and a random sojourn time $\tau_k \geq \sigma_k$.[1] In order to formulate the scheduler design problem, we introduce the arrival profiles, the service profiles, the system dynamics, and the design objectives below.

Arrival profiles. We represent the set of arriving jobs as a marked point process $\{(a_k; \sigma_k, \tau_k)\}_{k\in\mathcal{V}}$ in $\mathbb{R}_+ \times S$, where the arrival times $a_k \in \mathbb{R}_+$ are the set of points, and the service requirements $(\sigma_k, \tau_k) \in S$ are the set of marks. We assume that the marked point process is a stationary independently marked Poisson Point Process, which is defined by an intensity function Λ on $\mathbb{R}_+$ and a mark density measure $f(\sigma, \tau)$ on S [1]. This also implies that $\{(a_k; \sigma_k, \tau_k)\}_{k\in\mathcal{V}}$ is a Poisson point process on $\mathbb{R}_+ \times S$ with an intensity function $\Lambda f(\sigma, \tau)$. Intuitively, $\Lambda \int_A f(\sigma, \tau) d\sigma d\tau$ is the rate at which jobs with service requirement $(\sigma, \tau) \in A \subset S$ arrive. We additionally assume that S is bounded, and $S \subset \{(\sigma, \tau) : \tau \geq \sigma \geq 0\}$.

Service profiles. The service system works on each job $k \in \mathcal{V}$ with a *service rate* $r_k(t) \geq 0$. To meet the service demand of job k, its service rate must satisfy

$$\int_{a_k}^{\infty} r_k(t) dt = \sigma_k, \qquad k \in \mathcal{V}. \tag{1}$$

Moreover, the service rate can take non-zero values only when the job sojourns in the system, *i.e.*, $r_k(t) = 0$ for any $t \notin [a_k, a_k + \tau_k)$. Without loss of generality, $r_k(t) = 1$ is assumed to be the maximum rate: that is, $r_k(t)$ can take any values in $[0, 1]$, and $r_k < 1$ corresponds to throttling down service speed at the expense of prolonging job completion times. The above sojourn time and maximum rate constraints can be jointly written as

$$0 \leq r_k(t) \leq \mathbf{1}_{\{t\in[a_k, a_k+\tau_k)\}}, \tag{2}$$

where $\mathbf{1}_A$ denotes the indicator function for an event A. The service capacity is defined by

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

which is associated with the instantaneous resource consumption of the service system.

System dynamics. At each time $t \in \mathbb{R}_+$, job k has a remaining demand $x_k(t) = \sigma_k - \int_{a_k}^{t} r_k(h) dh$ and a remaining time $y_k(t) = a_k + \tau_k - t$. The set of remaining jobs in the system can be considered as a point process $\{(x_k(t), y_k(t))\}_k$ in $\mathbb{R}^2$, where the first coordinate

[1] The condition $\tau_k \geq \sigma_k$ requires each job $k \in \mathcal{V}$ to have a service demand σ_k that is no more than the maximum service that can be provided within its sojourn time τ_k.

(x) represents the remaining demand and the second coordinate (y) represents the remaining time. At time t, each point (job) has velocity $-r_k(t)$ in the direction of x-coordinate and velocity -1 in the direction of y-coordinate.

Scheduling algorithms. An online scheduling algorithm decides the service rates in real-time without using the future job arrival information. For scalability, we additionally restrict our attention to the following form of *distributed* algorithms which decide the service rate of a job only using its own information:

$$r_k(t) = u(x_k(t), y_k(t)) \geq 0. \tag{3}$$

Here, $u : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}_+$ is a deterministic function of the remaining demand $x_k(t)$ and the remaining time $y_k(t)$ of each job k at time t. The policy u also uniquely determines the vector field in the space of the point process $\{(x_k(t), y_k(t))\}_k$, which in turn defines the velocity $(-u(x, y), -1)$ of points (jobs) at (x, y) (see Fig 1).

Under any policy of the form (3), the set of jobs remaining in the system converges to a stationary distribution. This stationary distribution is a spatial Poisson point process with an intensity function $\lambda(x, y)$ satisfying

$$0 = \frac{\partial}{\partial x}(\lambda(x, y)u(x, y)) + \frac{\partial}{\partial y}\lambda(x, y) + \Lambda f(x, y), \tag{4}$$

where x is the remaining demand and y is the remaining time. Because the remaining job distribution converges to a stationary distribution, $P(t)$ also converges to a stationary distribution. [2]

Design objectives. We consider minimizing service capacity variability for the settings with hard 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:

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

where $\text{Var}(P)$ is a functional of u and $\lambda(\sigma, \tau)$ that satisfies (4). The optimization problem (5) has demand constraints as in (1) and deadline constraints as in (2). The constraint (3) restricts the optimization variable u to be distributed.

In the case of *soft demand constraints*, we relax the demand requirements (1) and penalize the amount of unsatisfied demands with a unit cost δ. In this setting, we consider balancing the service capacity variance and the expected cost for unsatisfied demands:

$$\underset{u:(2)(3)(4)}{\text{minimize}} \quad \text{Var}(P) + \mathbb{E}[\delta U], \tag{6}$$

where $U(t) = \sum_{k\in\mathcal{V}:a_k+\tau_k=t} x_k(t)$ is the total amount of remaining demands for jobs departing at time t.

In the case of *soft deadline constraints*, we relax the deadline requirements (2) and penalize deadline extensions with a unit cost ϵ. Let $\hat{\tau}_k$ be the actual sojourn time of job $k \in \mathcal{V}$, *i.e.*,

$$0 \leq r_k(t) \leq \mathbf{1}_{\{t\in[a_k, a_k+\hat{\tau}_k)\}}. \tag{7}$$

So $\hat{\tau}_k - \tau_k$ is the duration of deadline extension, and let

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

be the total duration of deadline extensions for jobs departing at time t. We consider balancing the service capacity variance and the expected cost for deadline extensions:

$$\underset{u:(1)(3)(4)(7)}{\text{minimize}} \quad \text{Var}(P) + \mathbb{E}[\epsilon W]. \tag{8}$$

In the case of *soft demand and deadline constraints*, we relax both the demand requirements (1) and the deadline requirements (2). The system needs to pay a cost of δ for each unit of unsatisfied demands and a cost of ϵ for each unit of deadline extensions. In this setting, we consider balancing the service capacity variance, the expected cost for unsatisfied demands, and the expected cost for deadline extensions:

$$\underset{u:(3)(4)(7)}{\text{minimize}} \quad \text{Var}(P) + \mathbb{E}[\delta U] + \mathbb{E}[\epsilon W]. \tag{9}$$

Generalizing above cases, we consider the case where the unit costs for unsatisfied demands and deadline extensions are heterogeneous among jobs. Let δ_k be the unit cost for the unsatisfied demand of job $k \in \mathcal{V}$, and ϵ_k be the unit cost for its deadline extension. The set of jobs is assumed to be a independently marked Poisson point process $\{(a_k; \sigma_k, \tau_k, \delta_k, \epsilon_k)\}_{k\in\mathcal{V}}$, where the unit costs $(\delta_k, \epsilon_k) \in \mathbb{R}^2_+$ are the additional marks of jobs. We assume that (δ_k, ϵ_k) are identically distributed random variables with a joint density measure $f(\delta)f(\epsilon)$ (hence independent from each other as well) and are also statistically independent from $(a_k; \sigma_k, \tau_k)$. To account for the heterogeneous costs, we consider scheduling policies of the form

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

and the optimization problem

$$\min_{\bar{u}:(7)(10)} \text{Var}(P(t)) + \mathbb{E}\left[\sum_{\substack{k\in\mathcal{V}:\\ a_k+\hat{\tau}_k=t}} \delta_k x_k(t)\right] + \mathbb{E}\left[\sum_{\substack{k\in\mathcal{V}:\\ a_k+\hat{\tau}_k=t}} \epsilon_k(\hat{\tau}_k - \tau_k)\right]. \tag{11}$$

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, and resource allocations in the Internet of Things. Note that we are not modeling a specific application, rather we are exploring the trade-offs in a simple, general model.

However, to highlight the connection to our motivating examples, consider first the case of electric vehicle charging [17]. 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 provides vehicle k with a charging rate of $r_k(t)$ by drawing $P(t) = \sum_{k\in\mathcal{V}} r_k(t)$ amount of power from the grid. When doing so, a stable resource usage is highly desirable because fluctuations and large peaks in $P(t)$ can lead to a high peak charge or strain the grid. Moreover, a predictable resource use 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.

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

[2] In this paper, we use the following notation: $\mathbb{E}[P]$ and $\text{Var}(P)$ represent the stationary mean and variance of a stochastic process $\{P(t)\}_{t\in\mathbb{R}_+}$, while $\mathbb{E}[P(t)]$ and $\text{Var}(P(t))$ represent the instantaneous mean and variance of $P(t)$ at time t.

$P(t) = \sum_{k \in \mathcal{V}} r_k(t)$ number of computers (or amount of power). Given a good estimate of the future resource use, a cloud content provider can reserve resources through a long-term contract, whose price is lower and less volatile than that of a short term contract. This motivates its scheduling algorithm to achieve a predictable resource use.

3 OPTIMAL DISTRIBUTED ALGORITHMS

In this section, we characterize optimal distributed scheduling policies in a wide range of settings, starting with the simplest and moving toward the most complex. To begin, we focus on strict service requirements and show that Exact Scheduling minimizes the stationary variance of the service capacity (Section 3.1). Relaxing the demand requirements, we show that a variation of Exact Scheduling minimizes the weighted sum of both the stationary variance of the service capacity and the penalty for unsatisfied demand (Section 3.2). Relaxing the deadline requirements, we show that a different variation of Exact Scheduling minimizes the weighted sum of both the stationary variance of the service capacity and the penalty for demand extension (Section 3.2). Finally, we consider the case when both the demand and deadline requirements are relaxed (Section 3.4) and show that the optimal policy becomes significantly more complex in this case. However, note that all the optimal algorithms we identify are in closed-form, and thus provide clear interpretations and insights regarding the optimal trade-offs between reducing service capacity variability, satisfying the demands, and meeting deadlines. Moreover, it is interesting that the minimum service capacity variance is achieved by these simple algorithms, all of which are extremely scalable and easy to implement.

3.1 Strict demand and deadline requirements

We first consider the case of strict service requirements and show a closed-form formula of the algorithm that minimizes the stationary variance $\mathrm{Var}(P)$. To do so, it is worth noting that peaks in service rate amplifies the uncertainties in the future arrivals, which in turn produces large variance in $P(t) = \sum_k r_k(t) = \sum_k u(x_k(t), y_k(t))$. In order to minimize peaks subject to strict service requirements, one can consider using a flat service rate, which is achieved by the scheduling policy

$$u(x,y) = \begin{cases} \dfrac{x}{y}, & \text{if } y > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

This policy is known as Exact Scheduling and works by finishing all jobs *exactly* at their deadlines using constant service rates (Figure 1). It is also highly scalable because it is distributed and asynchronous, and it does not require much computation or memory use. Although existing literature has analyzed its performance in various settings [8, 13, 19, 20], optimality guarantees have been difficult to obtain. In this section, we show that Exact Scheduling minimizes the variance of service capacity under time-homogeneous job arrivals and strict demand constraints.

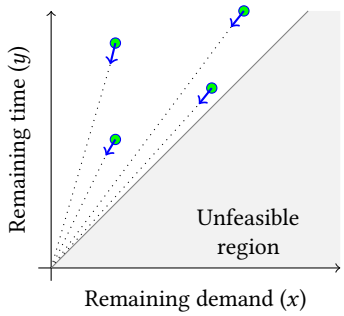

Figure 1: *Exact scheduling depicted in the space of remaining demand x and remaining time y.*

Theorem 3.1. *Exact Scheduling* (12) *is the optimal solution of* (5) *and achieves the optimal value*

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

Theorem 3.1 shows the achievable performance improvement by performing distributed service capacity control. If no control is applied, then $r_k(t) = \mathbf{1}_{[a_k, a_k+\sigma_k]}(t)$, and the stationary mean and variance of $P(t)$ is $\mathbb{E}(P) = \mathrm{Var}(P) = \Lambda \mathbb{E}[\sigma]$. By performing a distributed service capacity control, the stationary variance can be reduced by

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

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

3.2 Soft demand requirements

In this section, we relax the strict service requirements and characterize the optimal algorithm under soft demand constraints. Specifically, we consider the setting when the system needs to pay a cost δ for each unit of unsatisfied demands. When this unit cost is sufficiently large, we recover the case of strict service requirements. The optimal algorithm we identify is a generalization of Exact Scheduling:

$$u(x,y) = \begin{cases} \dfrac{x}{y}, & \text{if } \dfrac{x}{y} \le \dfrac{\delta}{2} \text{ and } y > 0, \\ \dfrac{\delta}{2}, & \text{if } \dfrac{x}{y} > \dfrac{\delta}{2} \text{ and } y > 0, \\ 0, & \textit{otherwise.} \end{cases} \tag{13}$$

We call (13) *Rate-limited Exact Scheduling*. This policy essentially sets $\delta/2$ to be the upper bound on service rates. Under this policy, job k receives its full service demand if $\sigma_k \le \delta\tau_k/2$ but otherwise is provided with the partial service demand of $\delta\tau_k/2$. To the best of our knowledge, this algorithm has not been proposed in the existing literature.

Theorem 3.2. *Rate-limited Exact Scheduling* (13) *is the optimal solution of* (6) *and achieves the optimal value*

$$\Lambda \mathbb{E}\left[\frac{\sigma^2}{\tau}\mathbf{1}_{\left\{\frac{\sigma}{\tau} \le \frac{\delta}{2}\right\}} + \delta\left(\sigma - \frac{\delta\tau}{4}\right)\mathbf{1}_{\left\{\frac{\sigma}{\tau} > \frac{\delta}{2}\right\}}\right]. \tag{14}$$

Theorem 3.2 shows the performance improvement gained by relaxing the demand requirements. If some demands do not have to

be satisfied, the stationary variance can be reduced from $\mathrm{Var}(P) = \mathbb{E}\left[\sigma^2/\tau\right]$ to (14) when the service rate threshold is set to its optimal value $\delta/2$. We prove Theorem 3.2 in the Appendix.

3.3 Soft deadline requirements

The previous section shows the optimal algorithm under soft demand requirements. In this section, we characterize the optimal distributed algorithm under soft deadline requirements. Specifically, we consider the setting when the system needs to pay a cost ϵ for each unit of deadline extensions. When the unit cost is sufficiently large, this setting recovers the case of strict deadline requirements. The resulting optimal algorithm is again a generalization of Exact Scheduling:

$$u(x,y) = \begin{cases} \dfrac{x}{y}, & \text{if } \dfrac{x}{y} \le \sqrt{\epsilon} \text{ and } y > 0, \\ \sqrt{\epsilon}\ \mathbf{1}_{\{x>0\}}, & \text{otherwise.} \end{cases} \tag{15}$$

We call (15) *Deadline-extended Exact Scheduling*. This policy essentially sets an upper bound $\sqrt{\epsilon}$ to service rates. Under this policy, the deadline of job k is extended when $\sigma_k > \sqrt{\epsilon}\tau_k$.

Theorem 3.3. *Deadline-extended Exact Scheduling* (15) *is the optimal solution of* (6) *and achieves the optimal value*

$$\Lambda\mathbb{E}\left[\frac{\sigma^2}{\tau}\mathbf{1}_{\left\{\frac{\sigma}{\tau}\le\sqrt{\epsilon}\right\}} + \left(2\sqrt{\epsilon}\sigma - \epsilon\tau\right)\mathbf{1}_{\left\{\frac{\sigma}{\tau}>\sqrt{\epsilon}\right\}}\right]. \tag{16}$$

Theorem 3.3 shows the performance improvement by relaxing the deadline requirements. If all deadline must be satisfied, then $\mathrm{Var}(P) = \Lambda\mathbb{E}\left[\sigma^2/\tau\right]$ is the minimum stationary variance achievable. If some deadlines do not have to be satisfied, the stationary variance can be further reduced at the expense of paying a penalty for deadline extensions. The service rate threshold $\sqrt{\epsilon}$ strikes the optimal balance between minimizing $\mathrm{Var}(P)$ and minimizing $\mathbb{E}[\epsilon W]$. We proof Theorem 3.3 in the Appendix.

3.4 Soft demand and deadline requirements

In this section, we consider relaxing both demand and deadline requirements simultaneously and characterize the optimal distributed algorithm. Specifically, we consider the setting when the system needs to pay a cost δ for each unit of demand extensions and a cost ϵ for each unit of deadline extensions. This setting recovers all previous settings as special cases.

Recall from previous sections that, under soft demand requirements, the optimal policy uses a constant service rate and reject partial demand requests if $\sigma/\tau > \delta/2$. Meanwhile, under soft soft deadline requirements, the optimal policy uses a constant service rate and extends the deadline if $\sigma/\tau > \sqrt{\epsilon}$. These two special cases suggest that, under soft demand and deadline requirements, a constant service rate combined with demand rejection and deadline extension may work well. This is indeed the case, as formalized below.

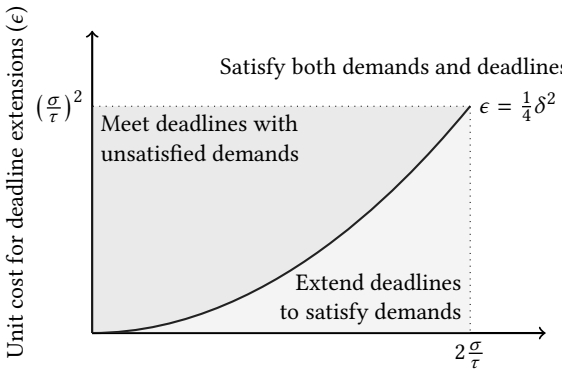

Figure 2: The decision space of the optimal policy for (9)**. For job profiles with a service demand σ, a sojourn time τ, and costs (δ, ϵ), the optimal policy performs either one of the following using constant service rates: satisfy both the job demand and deadline (white region), meet deadlines with unsatisfied demand (dark gray region), or satisfy the demand by extending the deadline (light gray region).**

Theorem 3.4. *The optimal solution of* (9) *is*

$$u(x,y) = \begin{cases} \dfrac{x}{y}, & \text{if } y > 0 \text{ and } \dfrac{x}{y} \le \min\left\{\dfrac{\delta}{2}, \sqrt{\epsilon}\right\}, \\ \dfrac{\delta}{2}, & \text{if } y > 0 \text{ and } \dfrac{x}{y} > \dfrac{\delta}{2} \text{ and } \dfrac{\delta}{2} \le \sqrt{\epsilon}, \\ \sqrt{\epsilon}\ \mathbf{1}_{\{x>0\}}, & \text{otherwise,} \end{cases} \tag{17}$$

and it achieves the optimal value

$$\Lambda\mathbb{E}\left[\frac{\sigma^2}{\tau}\mathbf{1}_{\left\{\frac{\sigma}{\tau}\le\min\left\{\frac{\delta}{2},\sqrt{\epsilon}\right\}\right\}} + \delta\left(\sqrt{\epsilon} - \frac{\delta\tau}{4}\right)\mathbf{1}_{\left\{\frac{\sigma}{\tau}>\frac{\delta}{2}\ge\sqrt{\epsilon}\right\}} + \left(2\sqrt{\epsilon}\sigma - \epsilon\tau\right)\mathbf{1}_{\left\{\frac{\sigma}{\tau}>\sqrt{\epsilon}>\frac{\delta}{2}\right\}}\right]. \tag{18}$$

We prove Theorem 3.4 in the Appendix. Theorem 3.4 shows when one should extend the deadline to satisfy the demand or let the job depart at its deadline with unsatisfied demands. The resulting optimal design space is shown in Figure 2, yielding the optimal policy (17). We summarize the strategy of (17) as follows:

- *High penalty regime.* For job profiles (σ, τ) satisfying $\delta/2 > \sigma/\tau$ or $\sqrt{\epsilon} > \sigma/\tau$ (outside of the colored rectangle in Figure 2), both its deadline and demand should be satisfied.
- *Low demand penalty regime.* When the unit cost for unsatisfied demands are comparatively smaller than that of deadline extension $\delta/2 \le \sqrt{\epsilon}$, for each job profile (σ, τ) satisfying $\delta/2 < \sigma/\tau, \sqrt{\epsilon} < \sigma/\tau$ (inside of the colored rectangle), its deadline should be satisfied.
- *Low deadline extensions penalty regime.* When the unit cost for deadline extension are comparatively smaller than that of unsatisfied demands $\delta/2 > \sqrt{\epsilon}$, for job profiles (σ, τ) satisfying $\delta/2 < \sigma/\tau, \sqrt{\epsilon} < \sigma/\tau$ (inside of the colored rectangle), its demand should be satisfied.

The above discussion highlights that (17) generalizes the optimal algorithms in Section 3.1-3.3, and we call (17) *Generalized Exact Scheduling*. Moreover, Generalized Exact Scheduling is also optimal for a more general problem (11), when the unit costs for unsatisfied demands and deadline extensions are allowed to be heterogeneous.

Corollary 3.5. *The optimal solution of* (11) *is*

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

4 PERFORMANCE BOUNDS

The focus of this work is on distributed algorithms, due to the importance of the algorithms being implementable in large-scale service systems. Given this focus, it is important to understand how much performance degradation is incurred due to restricting ourselves to distributed algorithms. To characterize the performance degradation, we compare the optimal distributed algorithm with the optimal centralized algorithms in this section by using both theoretical bounds and numerical comparisons. Specifically, we first provide an upper bound on the performance degradation. Then, we compare the optimal distributed online algorithms with existing centralized or offline algorithms using real Electric Vehicle charging instances [17].

Analytic bounds. To derive competitive-ratio-like bounds on the performance of optimal distributed policies, we first define centralized (online) policies and then bound their achievable performance. Then we compare this to performance bounds on the optimal distributed policies.

The class of centralized algorithms we consider is of the form

$$r_k(t) = w(k,t,A_t), \qquad \forall k \in \mathcal{V}, \tag{19}$$

where $A_t = \{(a_k, \sigma_k, x_k(t), y_k(t)) : a_k \le t\}$ is the set that contains the information of jobs arriving before t, and $w(k,t,\cdot)$ is a deterministic mapping from A_t to a service rate $r_k(t)$.

Lemma 4.1. *Under any centralized policy of the form* (19)*, the stationary variance of $P(t)$ is lower-bounded by*

$$\mathrm{Var}(P) \ge \frac{\Lambda^2 \mathbb{E}[\sigma^2]^2}{4\mathrm{Var}(X)},$$

where $X(t)$ is the total amount of remaining service demand of jobs arriving before t.

Lemma 4.1 characterizes the trade-off between achieving a small variance of $X(t)$ and achieving a small variance of $P(t)$. An immediate consequence of Lemma 4.1 is a competitive-ratio-like bound that compares Exact Scheduling (12) and the best centralized algorithm having the same $\mathrm{Var}(X)$ as Exact Scheduling. In particular, plugging in the stationary variance of X under Exact Schedule,

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

we obtain the following corollary.

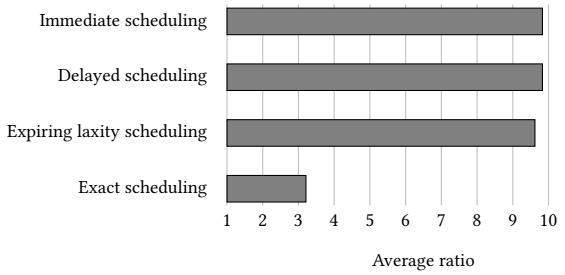

Figure 3: Comparison of algorithms under strict demand constraints. The competitive ratio of each algorithm is computed by the empirical $\mathrm{Var}(P)$ of the algorithm divided by the empirical $\mathrm{Var}(P)$ of the optimal centralized offline algorithm averaged over all instances.

Corollary 4.2. *Let $\mathrm{Var}(P^*)$ be the minimum stationary variance attainable by any centralized algorithm* (19) *with the same level of $\mathrm{Var}(X)$ as Exact Scheduling. Then, the stationary variance $P(t)$ that is attained by Exact Scheduling* (12) *satisfies*

$$\mathrm{Var}(P) \le \frac{\mathbb{E}[\sigma^2/\tau]\mathbb{E}[\sigma^2\tau]}{\mathbb{E}[\sigma^2]^2}\mathrm{Var}(P^*), \tag{20}$$

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

Corollary 4.2 bounds the ratio of $\mathrm{Var}(P)$, achievable by Exact Scheduling (the optimal distributed algorithm), and $\mathrm{Var}(P^*)$, achievable by any centralized algorithms. When the sojourn time τ is a deterministic random variable, (20) reduces to $\mathrm{Var}(P) \le \mathrm{Var}(P^*)$, implying that distributed algorithms can perform equally well compared to the centralized algorithms having the same $\mathrm{Var}(X)$. One such case is when service demands and sojourn times are deterministic variables, and the service demand of each job equals its sojourn time (arrival times a are random). In this case, due to the demand constraints (1) and the deadline constraints (2), $r_k(t) = \mathbf{1}_{\{t \in [a_k, a_k+\tau_k)\}}$ is trivially optimal both among centralized policies and among distributed policies.

Empirical performance. In order to further evaluate the performance of Exact Scheduling, we test it using data from an Electric Vehicle Charging Testbed [18] and compare the performance with existing scheduling algorithms. We employ a trace-driven simulation on a total of 92 charging instances from the testbed data in [17]. Each instance contains a set of jobs that are requested within a day. We compute the ratios between the empirical variance achieved by a few online algorithms and the empirical variance by the optimal centralized offline algorithm for all instances. The algorithms tested are Immediate scheduling ($u(x,y) = \mathbf{1}_{\{x>0\}}$), Delayed Scheduling ($u(x,y) = \mathbf{1}_{\{y \le x\}}$), Expiring Laxity (serving jobs with positive remaining laxity equally and serving jobs with zero laxity at its maximum rate [13]), and Exact Scheduling. For each algorithm tested, we plot the mean ratio in Figure 3. The results highlight significant performance gains compared to other distributed algorithms and competitive performance with the optimal centralized offline algorithm.

REFERENCES

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

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

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

[4] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. 1995. *Dynamic programming and optimal control.* Vol. 1. Athena scientific Belmont, MA.

[5] Partha P Bhattacharya and Anthony Ephremides. 1989. Optimal scheduling with strict deadlines. *IEEE Trans. Automat. Control* 34, 7 (1989), 721–728.

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

[7] Eric Boutin, Jaliya Ekanayake, Wei Lin, Bing Shi, Jingren Zhou, Zhengping Qian, Ming Wu, and Lidong Zhou. 2014. Apollo: Scalable and Coordinated Scheduling for Cloud-Scale Computing.. In *OSDI*, Vol. 14. 285–300.

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

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

[10] Niangjun Chen, Lingwen Gan, Steven H Low, and Adam Wierman. 2014. Distributional Analysis for Model Predictive Deferrable Load Control. In *Proc. of the IEEE 53rd annual Conference on Decision and Control.*

[11] Yiyu Chen, Amitayu Das, Wubi Qin, Anand Sivasubramaniam, Qian Wang, and Natarajan Gautam. 2005. Managing server energy and operational costs in hosting centers. In *ACM SIGMETRICS performance evaluation review*, Vol. 33. ACM, 303–314.

[12] Jeffrey Dean and Luiz André Barroso. 2013. The tail at scale. *Commun. ACM* 56, 2 (2013), 74–80.

[13] Andres Ferragut, Fernando Paganini, and Adam Wierman. 2017. Controlling the Variability of Capacity Allocations Using Service Deferrals. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)* 2, 3 (2017), 15.

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

[15] Anshul Gandhi. 2013. *Dynamic server provisioning for data center power management.* Ph.D. Dissertation. Carnegie Mellon University.

[16] J. Hong, X. Tan, and D. Towsley. 1989. A performance analysis of minimum laxity and earliest deadline scheduling in a real-time system. *IEEE Trans. Comput.* 38 (1989), 1736–1744.

[17] G. Lee, T. Lee, Z. Low, S. H. Low, and C. Ortega. 2016. Adaptive charging network for electric vehicles. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP).* 891–895. https://doi.org/10.1109/GlobalSIP.2016.7905971

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

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

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

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

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

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

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

[25] A. Nayyar, J. Taylor, A. Subramanian, K. Poolla, and P. Varaiya. 2013. Aggregate Flexibility of a Collection of Loads. In *Proc. of the 52nd IEEE Conference on Decision and Control.*

[26] Shivendra S Panwar and Don Towsley. 1988. On the optimality of the STE rule for multiple server queues that serve customers with deadlines. (1988).

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

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

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

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

[31] A. Subramanian, M.J. Garcia, D.S. Callaway, K. Poolla, and P. Varaiya. 2013. Real-Time Scheduling of Distributed Resources. *IEEE Transactions on Smart Grid* 4 (2013), 2122–2130.

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

Appendices

A PROOF OF THEOREM 3.1

To circumvent the complex constraints of (5), we first provide a lower bound on its optimal solution by relaxing the class of control policies into

$$r_k(t) = v(\sigma_k, \tau_k, y_k(t)) \qquad k \in \mathcal{V}, \tag{21}$$

and solve the optimization problem

$$\underset{v:(1)(2)(21)}{\text{minimize}} \quad \text{Var}(P(t)). \tag{22}$$

Because the constraint set of (5) is contained in the constraint set of (22), the optimal value of (22) lower-bounds that of (5). Therefore, to prove Theorem 3.1, it suffices to show that the optimal solution of (22) (given in the next lemma) is also achievable by a control policy of the form (12).

Lemma A.1. *The optimal solution of* (22) *is*

$$v(\sigma, \tau, y) = \frac{\sigma}{\tau}\mathbf{1}_{\{y>0\}}, \tag{23}$$

and it yields the optimal value $\text{Var}(P) = \mathbb{E}\left[\sigma^2/\tau\right]$.

To show Lemma A.1, we use the following Lemma.

Lemma A.2. *The mean and variance of* $P(t)$ *are given by*

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

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

Lemma A.2 is an immediate consequence of the fact that the steady state distribution is an independently marked Poisson Point Process, and thus its steady state characteristics can be recovered by appropriately integrating its mean measure [1]. Now we are ready to show Lemma A.1.

Proof. (Lemma A.1) The service demand constraints (1) are equivalent to

$$\int_0^\tau v(\sigma, y, \tau)dy = \sigma, \qquad (\sigma,\tau) \in S. \tag{24}$$

Combining (24) and Proposition A.2 , the optimization problem (22) can be rewritten into

$$\underset{v:(1)(21)(24)}{\text{minimize}} \int_{(\sigma,\tau)\in S}\int_0^\tau v(\sigma,\tau,y)^2\Lambda f(\sigma,\tau)dyd\sigma d\tau \tag{25}$$

The objective function of (25) satisfies

$$\begin{aligned}&\int_{(\sigma,\tau)\in S}\int_0^\tau v(\sigma,\tau,y)^2\Lambda f(\sigma,\tau)dyd\sigma d\tau\\&=\int_{(\sigma,\tau)\in S}\left\{\int_0^\tau v(\sigma,\tau,y)^2dy\right\}\Lambda f(\sigma,\tau)d\sigma d\tau\\&\geq\int_{(\sigma,\tau)\in S}\left\{\frac{\sigma^2}{\tau}\right\}\Lambda f(\sigma,\tau)d\sigma d\tau,\end{aligned}\tag{26}$$

where (26) is due to the Holder's inequality, *i.e.*, as $v(\sigma,\tau,y)\geq 0$ for any (σ,y,τ), we have

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

for any fixed (σ,τ). Alternatively, it can be verified that (26) can be attained with equality when v is given by (23). Therefore, (23) is the optimal solution of (22).

□

Theorem 3.1 is an immediate consequence of Lemma A.1. When a job with an arrival time a, a service demand σ, and a sojourn time τ is served according to (23), the optimal solution of (22), the ratio between its remaining demand $x(a+t)$ and remaining time $y(a+t)$ remains constant for any $t\in[a,a+\tau]$. Therefore, (23) can be realized using (12). This implies that the optimal solution (23) of (22) lies within the constraint set of (5). Because the optimal value of (22) is a lower bound on that of (5), it is also optimal for (5).

B PROOF OF THEOREM 3.2

Since the constraints of (6) are hard to solve, we first provide a lower bound on its optimal solution. Again, we consider the class of control policies representable by (21) and the optimization problem

$$\underset{v:(2)(4)(21)}{\text{minimize}}\quad \text{Var}(P(t))+\mathbb{E}[\delta U].\tag{27}$$

Because the constraint set of (27) contains that of (6), the optimal value of (27) lower-bounds that of (6). Therefore, to prove Theorem 3.2, it suffices to solve (27) (in the next lemma) and observe its optimal solution is representable by a control policy of the form (12).

Lemma B.1. *The optimal solution of* (27) *is*

$$v(\sigma,\tau,y)=\min\left\{\frac{\delta}{2},\ \frac{\sigma}{\tau}\right\}\mathbf{1}_{\{y>0\}},\tag{28}$$

and it achieves the optimal value (14).

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

$$\hat\sigma(\sigma,\tau)=\int_0^\tau v(\sigma,\tau,y)dy,\tag{29}$$

be the amount of service a job with a service demand σ and a sojourn time τ receives by its deadline. Then $\sigma-\hat\sigma(\sigma,\tau)$ is the amount of its unsatisfied demand. Additionally, $\hat\sigma(\sigma,\tau)$ satisfies

$$0\leq\hat\sigma(\sigma,\tau)\leq\sigma,\qquad\qquad\forall(\sigma,\tau)\in S.\tag{30}$$

Consequently, the stationary mean of U satisfies

$$\begin{aligned}\mathbb{E}[U]&=\lim_{t\to\infty}\mathbb{E}\left[\sum_{k\in\mathcal{V}:d_k=t}(\sigma_k-\hat\sigma(\sigma_k,\tau_k))\right]\\&=\int_{(\sigma,\tau)\in S}(\sigma-\hat\sigma(\sigma,\tau))\Lambda f(\sigma,\tau)d\sigma d\tau\end{aligned}\tag{31}$$

Then, we use (31) to rewrite (27) into

$$\begin{aligned}&\inf_{v:(2)(4)(21)}\ \text{Var}(P)+\delta\,\mathbb{E}[U]\\&=\inf_{\hat\sigma:(30)}\left[\inf_{v:(2)(4)(21)(29)}\text{Var}(P)\right.\qquad\qquad(32)\\&\qquad\left.+\delta\int_{(\sigma,\tau)\in S}(\sigma-\hat\sigma(\sigma,\tau))\Lambda f(\sigma,\tau)d\sigma d\tau\right]\\&=\inf_{\hat\sigma:(30)}\left[\left\{\inf_{v:(2)(4)(21)(29)}\text{Var}(P)\right\}\qquad(33)\right.\\&\qquad\left.+\delta\int_{(\sigma,\tau)\in S}(\sigma-\hat\sigma(\sigma,\tau))\Lambda f(\sigma,\tau)d\sigma d\tau\right].\end{aligned}$$

Equality (33) 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 (32) is not a function of v. From Lemma A.1, the first term of (33) admits the closed-form expression

$$\inf_{v:(2)(4)(21)(29)}\text{Var}(P)=\int_{(\sigma,s)\in S}\frac{\hat\sigma'(\sigma,\tau)^2}{\tau}\Lambda f(\sigma,\tau)d\sigma d\tau,\tag{34}$$

which is attained by

$$v(\sigma,\tau,y)=\frac{\hat\sigma}{\tau}.\tag{35}$$

Substitute (34) into (33) yields

$$\inf_{\hat\sigma:(30)}\int_{(\sigma,\tau)\in S}\left\{\frac{\hat\sigma'(\sigma,\tau)^2}{\tau}+\delta(\sigma-\hat\sigma'(\sigma,\tau))\right\}\Lambda f(\sigma,\tau)d\sigma d\tau,\tag{36}$$

where the optimization variable is now $\hat\sigma'$ instead of v. To derive a closed-form solution of (27), we can minimize the integrand in (36) point-wisely. By doing so, we observe that, for each $(\sigma,\tau)\in S$, a necessary and sufficient condition for optimality is

$$\hat\sigma(\sigma,\tau)=\arg\inf_{\hat\sigma:(30)}\frac{\hat\sigma(\sigma,\tau)^2}{\tau}+\delta(\sigma-\hat\sigma(\sigma,\tau))=\left\{\frac{\delta\tau}{2},\sigma\right\}.\tag{37}$$

Combining (35) and (37), we obtain (28) as the optimal control policy for (27). Substituting (28) into (36), we obtain its optimal value (14). □

□

Given Lemma B.1, Theorem 3.2 can be derived as follows. It can be verified that (28) can be attained using (13). This implies that the optimal solution of (27) lies within the constraint set of (6). Because the cost attained by (28) is a lower bound on the optimal value of (6), (28) is also optimal for (6).

C PROOF OF THEOREM 3.3

We first derive a lower bound on its optimal solution. Again, we consider the class of control policies representable by (21) and the optimization problem

$$\underset{v:(4)(7)(21)}{\text{minimize}} \quad \text{Var}(P(t)) + \mathbb{E}[\epsilon W]. \tag{38}$$

Because the optimal value of (38) lower-bounds that of (6), to prove Theorem 3.3, we can solve (38) (in the next lemma) and observe that its optimal solution is representable by a control policy of the form (3).

Lemma C.1. *The optimal solution of* (38) *is*

$$v(\sigma,\tau,y) = \begin{cases} \dfrac{\sigma}{\tau}\,\mathbf{1}_{\{y>0\}}, & \text{if } \dfrac{\sigma}{\tau} \le \sqrt{\epsilon}, \\ \sqrt{\epsilon}\,\mathbf{1}_{\left\{y>\tau-\frac{\sigma}{\sqrt{\epsilon}}\right\}}, & \text{otherwise} \end{cases}. \tag{39}$$

and it achieves the optimal value (16).

Proof. With a slight abuse of notation, let $\hat{\tau}(\sigma,\tau) \ge \tau$ denote the actual sojourn time for jobs having a service demand σ and a sojourn time τ. Then, the stationary mean of W satisfies

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

The optimization problem (38) can then be written into

$$\begin{aligned} &\inf_{v:(4)(7)(21)} \text{Var}(P) + \mathbb{E}[\epsilon W] \\ &= \inf_{\hat{\tau}\ge\tau} \left[\left\{ \inf_{v:(2)(4)(21)} \text{Var}(P) \right\} + \epsilon \int_{(\sigma,\tau)\in S} (\hat{\tau}(\sigma,\tau)-\tau)\Lambda f(\sigma,\tau)d\sigma d\tau \right] \qquad (40) \\ &= \inf_{\hat{\tau}\ge\tau} \int_{(\sigma,\tau)\in S} \left\{ \frac{\sigma^2}{\hat{\tau}} + \epsilon(\hat{\tau}(\sigma,\tau)-\tau) \right\} \Lambda f(\sigma,\tau)d\sigma d\tau, \qquad (41) \end{aligned}$$

where $\inf_{v:(2)(4)(21)} \text{Var}(P)$ in (40) is attained by

$$v(\sigma,\tau,y) = \frac{\sigma}{\hat{\tau}(\sigma,\tau)}. \tag{42}$$

The choice of $\hat{\tau}^*(\sigma,\tau)$ that is optimal for (8) is the point-wise maximum of the integrand of (41). So, $\hat{\tau}^*(\sigma,\tau)$ can be computed as

$$\hat{\sigma}(\sigma,\tau) = \arg\inf_{\hat{\sigma}:(30)} \frac{\hat{\sigma}(\sigma,\tau)^2}{\tau} + c(\sigma-\hat{\sigma}(\sigma,\tau)) = \left\{ \frac{c\tau}{2}, \sigma \right\}. \tag{43}$$

Combining (42) and (43), we obtain (39) as the closed-form solution of (38). □

Given Lemma C.1, we are now ready to prove Theorem 3.3.

Proof. (Theorem 3.3)

Recall that the optimal value of (38) lower-bounds that of (8). Therefore, if there is a policy of the form (3) that produces identical service rates to (39), it is also optimal for (8). Next, we show that Deadline-extended Exact Scheduling (15) satisfies the above description.

Given any job $k \in \mathcal{V}$ with $\sigma \le \tau\sqrt{\epsilon}$, both (15) and (39) yield 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 $\sigma < \sqrt{\epsilon}\tau$, (3) yields 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 (39), for any $y(t) > 0$, we have

$$\begin{aligned} \frac{x(t)}{y(t)} - \frac{\sigma}{\tau} &= \frac{\sigma - \sqrt{\epsilon}(t-a)}{\tau-(t-a)} - \frac{\sigma}{\tau} \\ &\ge \frac{(-\sqrt{\epsilon}+1)(t-a)}{\tau-(t-a)} \\ &\ge \frac{(-\sigma/\tau+1)(t-a)}{\tau-(t-a)} \\ &\ge 0, \end{aligned}$$

where the third inequality is due to $-\sqrt{\epsilon} \ge \sigma/\tau$. Thus, the policy (39) 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.

□

D PROOF OF THEOREM 3.4

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

$$\underset{v:(4)(7)(21)}{\text{minimize}} \quad \text{Var}(P(t)) + \mathbb{E}[\delta U] + \mathbb{E}[\epsilon W]. \tag{44}$$

The solution of (44) is given in the next lemma, which is then shown to be achievable under the constraints of (9) as well.

Lemma D.1. *The optimal solution of* (44) *is*

$$v(\sigma,\tau,y) = \begin{cases} \dfrac{\sigma}{\tau}\mathbf{1}_{\{y>0\}}, & \text{if } \dfrac{\sigma}{\tau} \le \min\left\{\dfrac{\delta}{2}, \sqrt{\epsilon}\right\}, \\ \dfrac{\delta}{2}\mathbf{1}_{\{y>0\}}, & \text{if } \dfrac{\sigma}{\tau} > \dfrac{\delta}{2} \text{ and } \dfrac{\delta}{2} \le \sqrt{\epsilon}, \\ \sqrt{\epsilon}\mathbf{1}_{\{x>0\}}, & \text{otherwise.} \end{cases} \tag{45}$$

and it achieves the optimal value (18).

Proof. Let $\hat{\sigma}(\sigma,\tau)$ denote the service demand for jobs having service demand σ and sojourn time τ, and let $\hat{\tau}(\sigma,\tau)$ denote the actual sojourn time for those jobs. The optimization problem (44) can be written into

$$\begin{aligned} &\inf_{v:(4)(7)(21)} \text{Var}(P(t)) + \mathbb{E}[\delta U] + \mathbb{E}[\epsilon W] \\ &= \inf_{\substack{\hat{\sigma}(\sigma,\tau)\ge\sigma \\ \hat{\tau}(\sigma,\tau)\ge\tau}} \Bigg[\inf_{v:(4)(7)(21)(29)} \text{Var}(P) \qquad (46) \\ &\quad + \int_{(\sigma,\tau)\in S} \{\delta(\sigma-\hat{\sigma}(\sigma,\tau)) + \epsilon(\hat{\tau}(\sigma,\tau)-\tau)\}\Lambda f(\sigma,\tau)d\sigma d\tau \Bigg] \\ &= \inf_{\substack{\hat{\sigma}(\sigma,\tau)\ge\sigma \\ \hat{\tau}(\sigma,\tau)\ge\tau}} \int_{(\sigma,\tau)\in S} \Bigg[\frac{\hat{\sigma}(\sigma,\tau)^2}{\hat{\tau}(\sigma,\tau)} \qquad (47) \\ &\quad + \delta(\sigma-\hat{\sigma}(\sigma,\tau)) + \epsilon(\hat{\tau}(\sigma,\tau)-\tau) \Bigg] \Lambda f(\sigma,\tau)d\sigma d\tau, \end{aligned}$$

where $\inf_{v:(4)(7)(21)(29)} \text{Var}(P)$ in (46) is attained by

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

The choice of $\hat{\sigma}^*(\sigma,\tau)$ and $\hat{\tau}^*(\sigma,\tau)$ for (44) is also the point-wise maximum of the integrand in (47), *i.e.*,

$$(\hat{\sigma}^*(\sigma,\tau),\hat{\tau}^*(\sigma,\tau)) = \arg \inf_{\substack{\hat{\sigma}(\sigma,\tau)\geq\sigma \\ \hat{\tau}(\sigma,\tau)\geq\tau}} \frac{\hat{\sigma}(\sigma,\tau)^2}{\hat{\tau}(\sigma,\tau)} + \delta(\sigma-\hat{\sigma}(\sigma,\tau)) + \epsilon(\hat{\tau}(\sigma,\tau)-\tau). \quad (48)$$

Next we derive the optimal solution $(\hat{\sigma}^*(\sigma,\tau),\hat{\tau}^*(\sigma,\tau))$ of (48). To this end, we first show that in the case of $\delta^2/4 \leq \epsilon$, we have $\hat{\tau}^*(\sigma,\tau)=\hat{\tau}$. Suppose not and $\hat{\tau}(\sigma,\tau)$ is optimal at some $\hat{\tau}>\tau$. The minimum of (48) subject to $\hat{\tau}(\sigma,\tau)=\hat{\tau}, \hat{\tau}\geq\tau$ can be computed as

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

When $\sigma\leq\delta\tau/2$, we

$$\begin{aligned} C(\hat{\tau})-C(\tau) &= \frac{\sigma^2}{\hat{\tau}}+\epsilon(\hat{\tau}-\tau)-\frac{\sigma^2}{\tau} \\ &= (\hat{\tau}-\tau)\left(\epsilon-\frac{\sigma^2}{\tau\hat{\tau}}\right) \\ &\geq (\hat{\tau}-\tau)\left\{\epsilon-\left(\frac{\delta\tau}{2}\right)^2\frac{1}{\tau\hat{\tau}}\right\} \quad (49) \\ &\geq (\hat{\tau}-\tau)\left\{\epsilon-\frac{\delta^2}{4}\right\} \quad (50) \\ &\geq 0, \quad (51) \end{aligned}$$

where (49) is due to $\sigma\leq\delta\tau/2$; (50) is due to $\hat{\tau}>\tau$; and (51) is due to $\delta^2/4\leq\epsilon$. When $\sigma\in(\delta\tau/2,\delta\hat{\tau}/2]$, we have

$$\begin{aligned} C(\hat{\tau})-C(\tau) &= \frac{\sigma^2}{\hat{\tau}}+\epsilon(\hat{\tau}-\tau)-\delta\left(\sigma-\frac{\delta\tau}{4}\right) \\ &\geq \epsilon(\hat{\tau}-\tau)+\left(\frac{\delta\tau}{2}\right)^2\frac{1}{\hat{\tau}}-\delta\frac{\delta\hat{\tau}}{2}+\frac{\delta^2\hat{\tau}}{4} \quad (52) \\ &\geq \epsilon(\hat{\tau}-\tau)+\frac{1}{2}\delta^2(\tau-\hat{\tau}) \quad (53) \\ &= (\hat{\tau}-\tau)\left\{\epsilon-\frac{\delta^2}{4}\right\} \\ &\geq 0, \quad (54) \end{aligned}$$

where (52) is due to $\sigma\leq\delta\tau/2$; (53) is due to $\hat{\tau}>\tau$; and (54) is due to $\delta^2/4\leq\epsilon$. When $\sigma>\delta\hat{\tau}/2$, we have

$$\begin{aligned} C(\hat{\tau})-C(\tau) &= \delta\left(\sigma-\frac{\delta\hat{\tau}}{4}\right)+\epsilon(\hat{\tau}-\tau)-\delta\left(\sigma-\frac{\delta\tau}{4}\right) \\ &= (\hat{\tau}-\tau)\left(\epsilon-\frac{\delta^2}{4}\right) \\ &\geq 0 \quad (55) \end{aligned}$$

where (55) is due to $\delta^2/4\leq\epsilon$. Since (51), (54), and (55) contradict with the assumption 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 B.1. 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 C.1. Finally, combining above, we obtain (45) as the closed-form solution of (44). □

Theorem 3.4 is an immediate consequence of Lemma D.1. Indeed, recall that the optimal value of (44) lower-bounds that of (9). Moreover, a policy of the form (3) can produce identical service rates to (45), so it is also optimal for (9).

E PROOF OF LEMMA 4.1

To solve $\inf_w L(w;\gamma)$, we first observe that

$$\begin{aligned} &\inf_w L(w;\gamma) \\ &= \inf_w \lim_{T\to\infty}\frac{1}{T}\int_0^T \mathrm{Var}(P(t))+\gamma(\mathrm{Var}(X(t))-D)dt \\ &\geq \inf_w \lim_{T\to\infty}\inf_w \frac{1}{T}\int_0^T \mathbb{E}[(P(t)-\bar{P})^2+\gamma((X(t)-\bar{X})^2-D)]dt \\ &= \lim_{T\to\infty}\inf_w \frac{1}{T}\int_0^T \mathbb{E}[(P(t)-\bar{P})^2+\gamma((X(t)-\bar{X})^2-D)]dt, \quad (56) \end{aligned}$$

where $\bar{P}$ and $\bar{X}$ are the stationary variance of $P(t)$ and $X(t)$ respectively. Now we represent the integral in (56) 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 $h=t_{n+1}-t_n, \forall n\in\mathbb{Z}_+$. Hence, the dynamics of $X(t_n)$ satisfies

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

where u is assumed to be constant during each sampling intervals. Then, (56) satisfies

$$\begin{aligned} &\lim_{T\to\infty}\inf_w \frac{1}{T}\int_0^T \mathbb{E}[(P(t)-\bar{P})^2+\gamma((X(t)-\bar{X})^2-D)]dt \\ &= \lim_{T\to\infty}\inf_w \lim_{h\to 0}\frac{1}{T}L_{h,\lceil T/h\rceil}(u;r)h-\gamma D \\ &= \lim_{T\to\infty}\lim_{h\to 0}\inf_w \frac{1}{T}L_{h,\lceil T/h\rceil}(u;r)h-\gamma D, \quad (57) \end{aligned}$$

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

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

To solve (57), we first consider the cost-to-go $J_n(X(t_n))$ for some $h>0$ and $N\in\mathbb{Z}_+$, *i.e.*,

$$J_n(X(t_n)) := \mathbb{E}\left[\gamma(X(t_N)-\bar{X})^2\right]+\sum_{k=n}^{N-1}\mathbb{E}\left[(P(t_k)-\bar{P})^2+\gamma(X(t_k)-\bar{X})^2\right]. \quad (58)$$

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

$$J_n(X(t_n)) = \mathbb{E}[p_n(X(t_n)-\bar{X})^2+\sum_{k=n}^{N-1}\mathbb{E}[p_{k+1}(A(t_n,h)-\bar{A})^2], \quad (59)$$

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

$$p_k = p_{k+1} - \frac{h^2 p_{k+1}^2}{h^2 p_{k+1} + 1} + \gamma, \qquad p_N = \gamma. \tag{60}$$

First, condition (59) holds for $n = N$. Second, if condition (59) holds for $n + 1$, then

$$\begin{aligned} J_n(X(t_n)) &= \inf_{P(t_n,h)} \mathbb{E}[(P(t_n) - \bar{P})^2 + \gamma(X(t_n) - \bar{X})^2 + J_{n+1}(X(t_{n+1}))] \\ &= \inf_{P(t_n,h)} \mathbb{E}[(P(t_n) - \bar{P})^2 + \gamma(X(t_n) - \bar{X})^2 \\ &\quad + p_{n+1}(X(t_n) + (A(t_n,h) - \bar{A}) - h(P(t_n) - \bar{P}))^2], \end{aligned} \tag{61}$$

where $\bar{A}_h$ are the stationary mean of $A(t_n, h)$, and $\bar{A}_h = h\bar{P}$ from Brumelle's formula. Expanding the last quadratic term in (61) and applying $\mathbb{E}[(A(t,h) - \bar{A})X(t_n)] = 0$, (61) can be written as

$$\begin{aligned} J_n(X(t_n)) =& (p_{n+1} + \gamma)(X(t_n) - \bar{X})^2 + \sum_{k=n}^{N} p_{k+1}\mathbb{E}(A(t_k,h) - \bar{A})^2] \\ &+ \inf_{P(t_n)} \{(1 + h^2 p_{n+1})(P(t_n,h) - \bar{P}_h)^2 \\ &- 2h\gamma p_{n+1}(X(t_n) - \bar{X})(P(t_n) - \bar{P})]. \end{aligned} \tag{62}$$

The minimum value of (62) is attained by

$$P(t_n,h) - \bar{P}_h = \frac{hp_n}{1 + h^2 p_n}(X(t_n) - \bar{X}), \tag{63}$$

and the optimal cost-to-go becomes (59), where p_n is defined by (60). As $N \to \infty$, p_k converges to a unique positive scalar

$$p := \lim_{N\to\infty} p_k = \frac{h^2\gamma + h\sqrt{\gamma}\sqrt{h^2\gamma + 4}}{2h^2}, \tag{64}$$

which is also a fixed point of (60) [4]. Taking the limit of $N \to \infty$ and $h \to 0$ for (63) and (64), the infimum of (57) is attained by

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

From (58), it also requires

$$\begin{aligned} \mathrm{Var}(P(t)) + \gamma \mathrm{Var}(X(t)) &= p\mathbb{E}[(A(t_k,h) - \bar{A})^2] \\ &= ph\Lambda\mathbb{E}[\sigma_0^2] \\ &= \sqrt{\gamma}\Lambda\mathbb{E}[\sigma_0^2]. \end{aligned}$$