Journal of Industrial and Management Optimization

https://www.aimspress.com/journal/jimo

JIMO, 22(10): 4901–4925.
DOI:10.3934/jimo.2026169
Received: 10 January 2026
Revised: 03 August 2026
Accepted: 25 August 2026
Published: 10 September 2026

Research article

A preference-based framework for the value of information in online scheduling: the single-machine now-or-later decision

Maria Zemzami[1], **Nhan-Quy Nguyen**[2,*], **Farouk Yalaoui**[2] **and Yassine Ouazene**[2]

[1] National Graduate School of Arts and Crafts, Mohamed V University, Avenue des Nations Unies, Rabat, Morocco

[2] Computer Sciences and Digital Society Laboratory (LIST3N), Chaire Connected Innovation, Université de Technologie de Troyes, 12 rue Marie Curie, CS 42060, Troyes 10004, France

* **Correspondence:** Email: nhan_quy.nguyen@utt.fr.

Abstract: Production scheduling increasingly requires decisions to be taken in real time, without complete knowledge of the jobs still to come. Competitive analysis provides worst-case guarantees for online algorithms but does not price the individual pieces of information a scheduler might acquire. This paper develops a preference-based framework that values information by the reduction it produces in the Pareto optimal action set.

We apply it to a *single, local, binary* decision epoch of the single machine problem $(1|online, r_j|\sum C_j)$: the *now-or-later* choice between dispatching the job in hand and idling for one announced arrival. It is a building block rather than a policy (we derive no competitive ratio), and this restriction is what makes closed-form values attainable. The optimal action depends on the two parameters only through a single comparison between the resequencing gain and the total waiting cost imposed on the queued jobs, so one bit of advice is necessary and sufficient. We obtain closed-form expected values when only one parameter is observed, together with an acquisition rule when forecasts are costly. Which forecast is worth more is governed by the upper tail of the gain distribution rather than by congestion alone: No crossover occurs for a power law of index $\alpha < 2$, whereas release-delay information prevails beyond a finite crossover for bounded or light-tailed gains. Evaluating the closed forms across five gain distributions places that crossover between 10 queued jobs and never.

Keywords: online scheduling; value of information; semi-online algorithms; competitive analysis; advice complexity; pareto optimality; heavy-tailed distributions
Mathematics Subject Classification: 90B35, 68W27

1. Introduction

Modern manufacturing systems place three demands on production scheduling that classical models handle poorly: an expanding set of environmental and social constraints, decisions that must be taken in real time in step with cyber-physical systems, and the absence of complete advance knowledge about future jobs [1]. Classical offline scheduling assumes perfect information and seeks global optimality, a paradigm that is often misaligned with the realities of modern production environments.

Online scheduling, where decisions are irrevocable and information is revealed gradually, offers a more realistic framework [2,3]. The standard performance metric is the *competitive ratio*: the worst-case ratio between the online algorithm's cost and the offline optimum. However, competitive analysis has limitations for practical applications. First, its worst-case guarantees can be pessimistic, ignoring typical scenarios where partial information (forecasts, bounds, historical data) is available at reasonable cost [4,5]. Second, competitive ratios compare complete algorithms but do not quantify the marginal value of individual pieces of information, which is precisely the quantity needed to justify forecasting investments.

This paper investigates the question: *How much is a given piece of information worth in online scheduling?* We develop a preference-based framework that quantifies information value through Pareto optimal action sets, and we apply it to one canonical local decision arising in the single machine problem $1|online, r_j|\sum C_j$, for which explicit formulas can be obtained.

1.1. Relation to existing paradigms

The transition from static, offline planning to dynamic, online decision-making represents a paradigm shift in scheduling theory. Introduced by Graham [6], online scheduling addresses scenarios where job attributes are revealed only upon arrival. This paradigm has been extensively adapted to various environments, including parallel machines [7,8] and job shops [9].

1.1.1. Competitive analysis: metric and limitations

The standard metric for online algorithms is *competitive analysis*, which measures the worst-case performance ratio against an omniscient offline adversary [2,3]. While this metric provides guarantees, it has been criticized for its "excessive pessimism" [10], often yielding theoretical bounds far looser than the performance observed in practice. Alternative metrics such as resource augmentation [11] and local competitiveness [12] have provided more nuance. Amortized analysis [13] is not an alternative metric; it is a proof technique that aggregates costs over a sequence of events, used to establish worst-case competitive guarantees without probabilistic average-case assumptions.

1.1.2. Semi-online scheduling

A body of work known collectively as *semi-online scheduling* has emerged to bridge the gap between total ignorance (online) and perfect knowledge (offline). These approaches assume that partial information is available, such as the total processing time, the largest job size, or a lookahead window [4,5]. A key finding in this domain is the distinction between *valuable* and *valueless* information: Certain types of partial knowledge allow for algorithms with strictly better competitive ratios (valuable), while others provide no worst-case advantage (valueless) [14].

1.1.3. Advice complexity

Parallel to semi-online scheduling, the field of *advice complexity* [15, 16] formalizes the "quantity" of information in bits required to achieve a target competitive ratio. A central question is to determine the minimum advice needed to attain a prescribed performance level (often optimality). Here, the *bit complexity* of an online problem denotes the minimum number of advice bits that an oracle must reveal to the algorithm in order to guarantee a target competitive ratio, regardless of what those bits actually encode. It therefore measures the raw *quantity* of information, abstracting away the operational meaning of any particular parameter. Our work differs by focusing on the *economic value* of specific operational parameters rather than this abstract bit complexity.

1.1.4. Value of information in operations research

In operations research more broadly, the value of information (VOI) is a standard concept in decision theory and stochastic programming, yet it has seldom been applied to worst-case online scheduling. More recently, the *learning-augmented* (or *algorithms-with-predictions*) paradigm has analyzed how possibly imperfect predictions can improve the competitive ratio of online algorithms, including for scheduling and caching [17–19]. These works characterize the worst-case benefit of predictions but do not assign an explicit economic value to individual operational parameters. The present paper addresses that gap by extending the preference-based framework of Borodin and El-Yaniv [2] so as to value information directly.

1.2. Contributions and paper organization

This paper quantifies the marginal value of specific pieces of information in online scheduling. We deliberately restrict the analytical core to a *single, deterministic, local, binary* now-or-later decision, used as a building block rather than a full-horizon policy; this stylized model is precisely what makes closed-form information-value expressions possible. We prove no competitive ratio and propose no online algorithm: The object of study is the worth of one forecast at one decision epoch. Within this scope, our contributions are threefold:

1. **A preference-based valuation framework.** We formalize *valuable information* as that which reduces the Pareto optimal action set, extending the preference-based framework of Borodin and El-Yaniv to the explicit valuation of information.
2. **Closed-form values and a comparison of information types.** For the single-machine now-or-later decision within $1|online, r_j|\sum C_j$, we derive explicit values for knowing the resequencing gain (Δ_p) versus the release time delay (Δ_r) and characterize when each dominates: The ranking is governed by the upper tail of the gain distribution, gap information prevailing at all queue sizes for a power law of index $\alpha < 2$ and losing to release-delay information beyond a finite crossover otherwise.
3. **Granularity and validation.** We prove that the optimal action depends on the two parameters only through a single scalar comparison, so that one bit of advice is necessary and sufficient for this local decision, and we confirm the analytical predictions by evaluating the closed-form expressions across five gap distributions spanning bounded, light-tailed, and genuinely power-law behavior.

The remainder of this paper is structured as follows. Section 2 develops the preference-based framework and Pareto optimal action sets. Section 3 analyzes the single machine now-or-later problem and derives

the value formulas. Section 4 compares the processing time gap versus release time delay information. Section 6 discusses managerial insights and practical applications. Section 7 concludes with future perspectives.

2. Preference-based framework for information valuation

2.1. Algorithmic decision problems

Following Borodin and El-Yaniv [2], we model the scheduling problem as a 3-tuple $(\mathcal{A}, \mathcal{S}, C)$, where $\mathcal{A}$ is a set of actions (scheduling decisions; possibly a continuum, as with the waiting durations of Section 3.3), $\mathcal{S}$ is a set of states (future scenarios), and $C(a, s)$ is the cost of action a in state s. In the offline setting, the state s is known before the action is chosen. In the online setting, the action must be selected before s is revealed.

2.2. Preference relations and Pareto optimality

We define a preference relation based on cost dominance. Given a subset of possible future scenarios $\mathcal{S}' \subseteq \mathcal{S}$, an action a is said to be *preferred* over action b (denoted $a \preccurlyeq_{\mathcal{S}'} b$) if it yields a lower or equal cost in all scenarios within that subset: $C(a, s) \leq C(b, s)$ for all $s \in \mathcal{S}'$. We adopt this cost-dominance relation as the modeling primitive for preferences; it is the standard partial order underlying competitive analysis [2], under which an action is preferred only if it is never more costly than the alternative across the admitted scenarios.

We say that b *strictly dominates* a on $\mathcal{S}'$ when $b \preccurlyeq_{\mathcal{S}'} a$ and the inequality is strict for at least one $s \in \mathcal{S}'$; that is, b is never worse and is somewhere better. This is the notion used throughout, and in particular in Proposition 3.2.

When knowledge is incomplete, no action typically dominates all the others. In such situations, we seek the *Pareto optimal action set* $\mathcal{A}^*(\mathcal{S}')$, which consists of all actions that are not strictly dominated by any other action given the available information $\mathcal{S}'$. Actions within this set are mutually indifferent, meaning that the choice between them depends on information not yet available.

2.3. Value of information

We quantify the VOI as the reduction in guaranteed cost. Let a_{mm} be the minimax action chosen under ignorance, defined formally in (2.2) below. The value $v(s_0)$ of knowing the true scenario s_0 is defined as the difference between the cost of the minimax action and the cost of the optimal action for that scenario:

$$v(s_0) = C(a_{\text{mm}}, s_0) - \min_{a \in \mathcal{A}} C(a, s_0). \tag{2.1}$$

This metric captures the performance gain achieved by eliminating the uncertainty that forced the adoption of the conservative a_{mm} strategy.

We define the minimax action formally as

$$a_{\text{mm}} \in \arg\min_{a \in \mathcal{A}} \max_{s \in \mathcal{S}} C(a, s). \tag{2.2}$$

This definition ties directly to the Pareto structure: By construction, $v(s_0) > 0$ exactly when a_{mm} is strictly dominated in state s_0, that is, when it is eliminated from the Pareto optimal set of the

singleton $\{s_0\}$. In this sense, *valuable* information is precisely information that removes a previously non-dominated action from the decision problem, which is the conceptual basis of our framework. We now apply the framework to a canonical local decision in online scheduling, quantifying the value of knowing release dates and processing times in a concrete setting.

3. Single machine scheduling: the now-or-later problem

We apply the preference framework to $1|online, r_j|\sum C_j$: schedule jobs on a single machine to minimize total completion time, where job j has processing time p_j and release date r_j. In this section, we analyze a deterministic one-step decision model at a fixed decision epoch, used as a local building block for broader online settings.

3.1. The single-machine problem: context and optimality

The offline problem $1|r_j|\sum C_j$ is strongly NP-hard [20]. When all jobs are available simultaneously ($r_j = 0$), the shortest processing time (SPT) rule is optimal [21].

At decision time t, let $\Omega(t)$ denote released but unscheduled jobs. The n jobs in $\Omega(t)$ are not yet committed and may still be reordered; the order in which they would actually be served is set by the dispatching rule in force, which may be SPT or a priority order inherited from earlier decisions or from external requirements such as due dates or customer classes. Once a job starts processing, it is not preempted in our setting. Our *myopic* subproblem conditions on a local decision epoch and asks which immediate action to take next, not how to optimize all future epochs at once. This local framing does not preclude strategic idling in general online scheduling models. By SPT optimality, if the scheduler decides to process immediately, the best myopic action is to schedule the shortest available job $j_0 = \arg\min_{j\in\Omega} p_j$.

3.2. The now-or-later decision: problem setup

Suppose that at time t, the machine is idle, and the queue $\Omega(t)$ contains n available (released but unscheduled) jobs, among which j_0 is the shortest, with processing time p_0. By the SPT argument of Section 3, j_0 is the candidate to be dispatched next. One future job x (processing time p_x) will arrive at $r_x > t$. We parameterize the decision by

$$\Delta_r = r_x - t \;>\; 0 \quad \text{(release time delay: how long until } x \text{ arrives)}, \tag{3.1}$$

$$\Delta_p = \textstyle\sum_{i<k}(p_i - p_x) \quad \text{(resequencing gain: the completion-time benefit of serving } x \text{ first)}. \tag{3.2}$$

Here Δ_r is the cost driver of waiting, while Δ_p is its potential benefit. The release delay is strictly positive because x arrives in the future.

The gain Δ_p is a resequencing gain, not a processing-time difference. We stress the interpretation of (3.2), which is used consistently throughout the paper. Let $j_0, j_1, \ldots, j_{n-1}$ denote the queued jobs in the order they would be served if j_0 were dispatched now, and let $k \geq 1$ be the number of them that x has advanced past when it is served first instead. Advancing x from position k to the front lowers its own completion time by $\sum_{i<k} p_i$ and raises that of each overtaken job by p_x, so the net reduction in $\sum_j C_j$ is exactly $\Delta_p = \sum_{i<k}(p_i - p_x)$, as stated. Two consequences matter for what follows:

- **Single displacement ($k = 1$).** Serving x immediately before j_0 rather than immediately after is an adjacent interchange, and (3.2) reduces to the literal processing-time gap $\Delta_p = p_0 - p_x$, which is bounded above by p_0. This is the case used in the worst-case construction of Section 3.3.
- **Multiple displacements ($k > 1$).** When the queued jobs carry priorities inherited from earlier decisions and a short arrival overtakes several of them, Δ_p accumulates k separate gaps and is therefore *not* bounded by p_0. This is why the stochastic analysis of Sections 3.4–4 and the experiments of Section 5 model Δ_p as an unbounded random variable on the real line.

The sign of Δ_p governs the advantage of waiting: $\Delta_p > 0$ means resequencing x ahead is favorable, $\Delta_p < 0$ means it is unfavorable, and $\Delta_p = 0$ signifies indifference. Throughout, "processing gap" is used as a shorthand for this resequencing gain.

Scope and assumptions. We study a single, deterministic decision epoch. The n jobs already in the queue $\Omega(t)$ are served in the order set by the dispatching rule in force, and the index k of (3.2) counts how many of them the arrival x overtakes under that rule. Under free reordering with x as the shortest job, $k = 1$; under priorities inherited from earlier decisions, k may be larger. The decision concerns only the highest-priority available job j_0: Dispatch it immediately, or wait for x and then re-optimize the currently considered set by SPT. No arrivals other than x are considered within this local comparison horizon. We adopt an additive parameterization (Δ_r, Δ_p) rather than a multiplicative one because it yields closed-form waiting conditions; the queue congestion enters through the factor $(n + 1)$ derived below. This is a one-step local model used to quantify marginal information value, not a full-horizon policy characterization.

Decision: Schedule j_0 immediately ("now"), or wait for x ("later")?

3.2.1. Cost-benefit analysis

Let G_{wait} denote the net benefit of waiting, i.e., the cost of dispatching now minus the cost of waiting for x and then re-optimizing by SPT:

$$G_{\text{wait}} = C(\text{dispatch now}) - C(\text{wait for } x, \text{ then re-optimize by SPT}). \tag{3.3}$$

We evaluate the two terms explicitly. Let n be the number of jobs in the queue $\Omega(t)$ at the decision time t.

If j_0 is *dispatched now*, it starts at t and imposes no global delay on the queue; when x arrives at $t + \Delta_r$, it is simply inserted into the remaining SPT schedule. We assume throughout that $\Delta_r \leq \sum_{j\in\Omega(t)} p_j$, so that x arrives before the queue is exhausted and this branch never idles; the assumption is comfortably satisfied in every regime we study, where Δ_r is small relative to a single processing time.

If the scheduler *waits*, the machine idles for a duration Δ_r. Consequently each of the n queued jobs *and* the arriving job x start Δ_r later than they otherwise would, so the total completion time increases by exactly $(n + 1)\,\Delta_r$. Against this delay, serving x first recovers precisely the resequencing gain Δ_p of (3.2). The remaining jobs are reordered optimally after the decision, so the delay penalty is uniform. Taking the difference gives

$$G_{\text{wait}} = \underbrace{\Delta_p}_{\text{gain from the shorter } x} - \underbrace{(n + 1)\,\Delta_r}_{\text{delay on the } n+1 \text{ jobs}}. \tag{3.4}$$

Here Δ_r refers to the arrival of the specific future job x; with multiple future arrivals, an indexed notation $\Delta_r^{(i)}$ would be needed, but for the single-job case, Δ_r is unambiguous.

Lemma 3.1 (Waiting condition)**.** *Waiting for job x reduces total completion time if and only if*

$$\Delta_p > (n+1) \times \Delta_r. \tag{3.5}$$

Proof. Waiting is beneficial if and only if $G_{\text{wait}} > 0$. The result follows directly from (3.4). □

Remark 3.1 (Intuitive interpretation)**.** The waiting condition (3.5) states that the *resequencing gain* Δ_p must exceed the *accumulated waiting cost* $(n+1)\Delta_r$. The factor $(n+1)$ reflects the local horizon used in the comparison: The n currently available jobs plus the arriving job x are all shifted by the waiting delay in the wait-first alternative. This makes the trade-off explicit: Larger queues ($n \uparrow$) or longer delays ($\Delta_r \uparrow$) require proportionally larger gains ($\Delta_p \uparrow$) to justify waiting.

3.3. Pareto optimal actions under ignorance

This subsection is confined to the single-displacement case $k = 1$ of Section 3.2, for which $\Delta_p = p_0 - p_x \leq p_0$: A worst-case agent commits to wait only as long as a gain it can be sure of recovering would justify, and the guaranteed gain of one adjacent interchange is at most p_0. The multiple-displacement case, in which Δ_p is unbounded, is treated stochastically in Section 3.4.

To characterize the decision space, we define three Boolean conditions. The primary condition determining whether waiting is beneficial is $\kappa_0 : \Delta_p > (n+1)\Delta_r$. We also identify two conditions implied by it: $\kappa_1 : \Delta_r < \frac{p_0}{n+1}$, which ensures the delay is bounded by a conservative threshold, and $\kappa_2 : \Delta_p > 0$, which simply states that the incoming job is shorter than the current one. We emphasize that κ_1 and κ_2 are *necessary but not sufficient*: They do not jointly determine the optimal action, which is governed by the single comparison κ_0 (see Proposition 3.5). With the recoverable gain of the single-displacement case bounded by p_0, the waiting cost $(n+1)\Delta_r$ exceeds any justifiable gain if $\Delta_r \geq \frac{p_0}{n+1}$. Thus κ_1 defines the "window of opportunity" for profitable waiting.

Remark 3.2. The condition κ_0 implies both κ_1 and κ_2. Since $\Delta_r > 0$ and $n \geq 0$, $\Delta_p > (n+1)\Delta_r$ implies $\Delta_p > 0$ (κ_2). Furthermore, under the single-displacement bound $\Delta_p \leq p_0$ adopted in this subsection, κ_0 implies $(n+1)\Delta_r < p_0$ or $\Delta_r < \frac{p_0}{n+1}$ (κ_1). Thus, κ_1 and κ_2 are necessary conditions for waiting to be beneficial. The converse fails: $\kappa_1 \wedge \kappa_2$ does not imply κ_0, since a positive but small gain Δ_p may still fall short of the waiting cost $(n+1)\Delta_r$.

Let $\Delta_r^{\max} = \frac{p_0}{n+1}$ denote the conservative waiting threshold. We consider two primary actions available to the decision-maker: a_0, which schedules j_0 immediately, and a_1, which waits up to the safe threshold $\Delta_r^{\max}$.

The regret matrix for these actions is shown in Table 1.

Table 1. Worst-case regret within each scenario, under ignorance (n jobs queued). The entry p_0 is attained when $\Delta_r \geq \Delta_r^{\max}$, so that a_1 idles for its full window and gains nothing; the other entries are exact.

Action	κ_0 (wait beneficial)	$\overline{\kappa_0}$ (wait detrimental)
a_0: Dispatch now	$G_{\text{wait}} = \Delta_p - (n+1)\Delta_r$	0
a_1: Wait $\Delta_r^{\max}$	0	p_0

Let a_w, for $w \geq 0$, denote the strategy that keeps the machine idle for up to w time units, serving x if it arrives within that window and dispatching j_0 otherwise; a_0 and a_1 are the members $w = 0$ and $w = \Delta_r^{\max}$ of this family. Evaluating the regret of a_w in the two scenarios as in Table 1 gives

$$C(a_w, \kappa_0) \;=\; (\Delta_p - (n+1)\Delta_r)\,\mathbb{1}\{\Delta_r > w\}, \qquad C(a_w, \overline{\kappa_0}) \;=\; (n+1)\,\min(w, \Delta_r), \tag{3.6}$$

the first because the gain is forfeited exactly when x arrives after the window has closed, the second because a detrimental wait costs the idle time it consumes. Setting $w = 0$ and $w = \Delta_r^{\max}$ recovers the two rows of Table 1. The first expression is non-increasing in w, and the second is non-decreasing, which is the tension the proposition resolves.

Proposition 3.2. *Under complete ignorance ($\mathcal{S} = \{\kappa_0, \overline{\kappa_0}\}$), the Pareto optimal action set is the whole continuum of waiting durations up to the conservative threshold,*

$$\mathcal{A}^*(\mathcal{S}) \;=\; \{\, a_w \;:\; 0 \leq w \leq \Delta_r^{\max} \,\},$$

of which a_0 and a_1 are the two extreme points. No member is dominated by any other, and every a_w with $w > \Delta_r^{\max}$ is dominated by a_1.

Proof. Take $0 \leq w < w' \leq \Delta_r^{\max}$. By (3.6), $a_{w'}$ is strictly better than a_w in the scenario κ_0 (it captures the arrivals with $w < \Delta_r \leq w'$, which a_w forfeits), while a_w is strictly better than $a_{w'}$ in the scenario $\overline{\kappa_0}$ (it consumes strictly less idle time, $(n+1)w < (n+1)w'$). Neither, therefore, dominates the other in the sense of Section 2, and this holds for every such pair; the whole family is mutually non-dominated. In particular, a_0 and a_1 realize the two extremes: a_0 is the unique minimizer of the regret under $\overline{\kappa_0}$ and a_1 the unique minimizer under κ_0.

For $w > \Delta_r^{\max}$, the extra waiting buys nothing: κ_0 implies $\Delta_r < \Delta_r^{\max}$ (Remark 3.2), so $C(a_w, \kappa_0) = C(a_1, \kappa_0) = 0$, whereas $C(a_w, \overline{\kappa_0}) = (n+1)w > p_0 = C(a_1, \overline{\kappa_0})$. Hence a_1 dominates a_w, and the Pareto set is exactly $\{a_w : 0 \leq w \leq \Delta_r^{\max}\}$. □

Remark 3.3. This is the concrete instance of the continuum of actions announced in Section 2. Under ignorance, the decision-maker is left with a one-parameter family of mutually indifferent waiting durations, and the role of information is precisely to collapse it. By Proposition 3.5, a single bit reduces this continuum to a single optimal action.

3.4. Value of partial information

From here on, we treat Δ_p as a general resequencing gain on the real line, no longer restricted to the single-displacement case of Section 3.2: When a short incoming job is resequenced ahead of several queued jobs, the completion-time benefit accumulates $k > 1$ gaps and may exceed a single processing time, so the relevant gain is unbounded above (and Section 5 accordingly uses unbounded gain distributions). Definition (2.1) measures the value of *fully* revealing the state. Whereas the construction of Sections 2 and 3.3 took a worst-case (minimax) stance, we now adopt an expectation viewpoint, natural when a prior over (Δ_r, Δ_p) is available; the no-information benchmark is accordingly the prior-mean rule introduced below. We quantify the value of observing only *one* of the two parameters, the unobserved one being replaced by its prior mean. This is the operationally relevant setting (a forecast may reveal the processing gap but not the exact arrival time, or vice versa), and it is precisely the model validated numerically in Section 5.

Let $\mu_r = \mathbb{E}[\Delta_r]$ and $\mu_p = \mathbb{E}[\Delta_p]$ denote the prior means. Throughout Sections 3.4–4, we assume that Δ_r and Δ_p are independent under the prior. This is what makes replacing an unobserved parameter by its unconditional mean the correct response to observing the other one; under dependence, the policies below would use $\mathbb{E}[\Delta_p \mid \Delta_r]$ and $\mathbb{E}[\Delta_r \mid \Delta_p]$ instead, and the value of each source would additionally reflect what it reveals about the other. Recall from Lemma 3.1 that waiting is ex-post optimal if and only if $\Delta_p > (n+1)\Delta_r$, and that, relative to dispatching now, the realized cost of waiting is $(n+1)\Delta_r - \Delta_p$ (a gain when negative). We compare three policies:

- **No information** ($a_\varnothing$): a fixed rule based on prior means, waiting if and only if $\mu_p > (n+1)\mu_r$;
- **Observe** Δ_r (a_{Δ_r}): wait if and only if $\mu_p > (n+1)\Delta_r$;
- **Observe** Δ_p (a_{Δ_p}): wait if and only if $\Delta_p > (n+1)\mu_r$.

The value of an information source is the expected reduction in cost relative to $a_\varnothing$. We keep the symbol ν for this quantity, but the benchmark has changed: In (2.1), it was the minimax action a_{mm}, whereas from here on it is the prior-mean rule $a_\varnothing$. The quantities $\mathbb{E}[\nu(\Delta_r)]$ and $\mathbb{E}[\nu(\Delta_p)]$ below are therefore *not* expectations of (2.1); they are its expectation-benchmark counterparts, and they measure the value of observing one parameter rather than of revealing the whole state. Write $(x)^+ = \max(0, x)$.

Proposition 3.3 (Value of observing Δ_r)**.** *The expected value of observing the release delay is*

$$\mathbb{E}[\nu(\Delta_r)] = \mathbb{E}_{\Delta_r}\big[(\mu_p - (n+1)\Delta_r)^+\big] - (\mu_p - (n+1)\mu_r)^+ \;\geq\; 0. \tag{3.7}$$

Proof. Conditional on Δ_r, the expected cost of waiting (over the unobserved Δ_p) is $(n+1)\Delta_r - \mu_p$ versus 0 for dispatching. Policy a_{Δ_r} selects the smaller, incurring conditional expected cost $\min(0,\ (n+1)\Delta_r - \mu_p) = -(\mu_p - (n+1)\Delta_r)^+$. Taking expectations over Δ_r gives $\mathbb{E}[\text{cost of } a_{\Delta_r}] = -\,\mathbb{E}[(\mu_p - (n+1)\Delta_r)^+]$. The fixed policy $a_\varnothing$ has expected cost $\min(0,\ (n+1)\mu_r - \mu_p) = -(\mu_p - (n+1)\mu_r)^+$. Subtracting yields (3.7). Non-negativity follows from Jensen's inequality applied to the convex map $\delta \mapsto (\mu_p - (n+1)\delta)^+$. □

Proposition 3.4 (Value of observing Δ_p)**.** *The expected value of observing the processing gap is*

$$\mathbb{E}[\nu(\Delta_p)] = \mathbb{E}_{\Delta_p}\big[(\Delta_p - (n+1)\mu_r)^+\big] - (\mu_p - (n+1)\mu_r)^+ \;\geq\; 0. \tag{3.8}$$

Proof. Symmetric to the previous proof: Conditional on Δ_p, policy a_{Δ_p} incurs expected cost $\min(0,\ (n+1)\mu_r - \Delta_p) = -(\Delta_p - (n+1)\mu_r)^+$. Subtracting the cost of $a_\varnothing$ and applying Jensen's inequality gives (3.8). □

Remark 3.4. Formulas (3.7)–(3.8) are of the classic "$\mathbb{E}[\max] - \max[\mathbb{E}]$" (expected value of perfect information) type, and they are genuinely *asymmetric*; observing Δ_r sharpens the wait threshold against the random delay, whereas observing Δ_p exploits the upper tail of the gap distribution. This asymmetry, absent from the worst-case decision (which depends only on the single comparison $\Delta_p \gtrless (n+1)\Delta_r$), is what drives the comparison in Section 4.

3.5. Information granularity and advice complexity

A key question in advice complexity theory is determining the minimum amount of information required to achieve optimality. Throughout this subsection, "optimal" refers to the *binary local action*

of Lemma 3.1 (dispatch j_0 now, or wait for x) and not to the expected-value magnitudes of Section 3.4, which vary continuously with the observed parameter. The statements below are deliberately confined to that single decision epoch; Section 5.7 discusses why they do not extend verbatim to a full-horizon policy.

Proposition 3.5 (Sufficiency of the joint threshold)**.** *The optimal action of Lemma 3.1 depends on* (Δ_r, Δ_p) *only through the sign of the scalar statistic*

$$G_{\text{wait}} = \Delta_p - (n+1)\,\Delta_r, \tag{3.9}$$

that is, through the comparison of the resequencing gain with the total waiting cost imposed on the $n+1$ *affected jobs. Consequently,*

1. *any information that determines the sign of* G_{wait} *is decision-equivalent to exact knowledge of both parameters; in particular, interval bounds* $\Delta_p \in [\underline{v}_p, \overline{v}_p]$ *and* $\Delta_r \in [\underline{v}_r, \overline{v}_r]$ *suffice as soon as* $\underline{v}_p > (n+1)\overline{v}_r$ *(wait) or* $\overline{v}_p \le (n+1)\underline{v}_r$ *(dispatch now);*
2. *separate thresholds on the two parameters do* not *suffice. The conditions* $\kappa_1 : \Delta_r < \frac{p_0}{n+1}$ *and* $\kappa_2 : \Delta_p > 0$ *of Section 3.3 are necessary for waiting to be beneficial but are not jointly sufficient, so knowing the position of each parameter relative to* $\tau_{\Delta_r} = \frac{p_0}{n+1}$ *and* $\tau_{\Delta_p} = 0$ *leaves the action undetermined in general.*

Proof. By (3.4), the cost difference between the two actions equals G_{wait}, so the optimal action is "wait" if $G_{\text{wait}} > 0$ and "dispatch now" otherwise, and nothing else about (Δ_r, Δ_p) enters. This proves the first claim; the two stated bound conditions are exactly those under which the sign of G_{wait} is constant over the rectangle $[\underline{v}_p, \overline{v}_p] \times [\underline{v}_r, \overline{v}_r]$, since G_{wait} is increasing in Δ_p and decreasing in Δ_r. For the second claim, it suffices to exhibit two instances satisfying $\kappa_1 \wedge \kappa_2$ with opposite optimal actions: With $n = 5$ and $p_0 = 10$, so that $\tau_{\Delta_r} = 10/6$, take $\Delta_r = 1.5 < \tau_{\Delta_r}$ in both, and $\Delta_p = 2 > 0$ in the first and $\Delta_p = 9.5 > 0$ in the second. Both satisfy κ_1 and κ_2, and both are realizable in the single-displacement case since $\Delta_p \le p_0 = 10$, yet $G_{\text{wait}} = 2 - 9 < 0$ in the first (dispatch now) and $G_{\text{wait}} = 9.5 - 9 > 0$ in the second (wait). □

This property implies that the "information granularity" required to make the local decision is extremely coarse: A single scalar comparison, rather than the two parameter values, is all that must be resolved.

Corollary 3.6 (Binary advice suffices for the local decision)**.** *For the binary now-or-later decision of Lemma 3.1, a single bit, namely the answer to the query "Is* $\Delta_p > (n+1)\Delta_r$*?", is necessary and sufficient to select the optimal action, and exact parameter values yield no better decision. Sufficiency is Proposition 3.5; necessity is immediate whenever both actions can be optimal, since the decision is binary. The result is stated for one decision epoch in isolation and does not assert that one bit per epoch suffices to drive an optimal full-horizon policy (Section 5.7).*

Example 3.1. Consider a system with $n = 5$ jobs in queue. Let $p_0 = 10$. A future job x arrives with delay $\Delta_r = 1.5$. The waiting cost threshold is $(n+1) \times \Delta_r = 6 \times 1.5 = 9$. If the resequencing gain is $\Delta_p = 2$, then $\Delta_p < (n+1) \times \Delta_r$. Binary advice would correctly indicate "Do not wait". Exact knowledge providing $\Delta_r = 1.5$ and $\Delta_p = 2$ yields the same decision. Note that the two *separate* tests both pass here: κ_1 reads $1.5 < 10/6$, and κ_2 reads $2 > 0$. A rule based on these two thresholds alone would therefore conclude that waiting is beneficial, which is incorrect. Only the joint comparison (3.9) settles the decision.

4. Comparative analysis: which information is more valuable?

4.1. Expected value under uncertainty

Propositions 3.3–3.4 provide $\mathbb{E}[\nu(\Delta_r)]$ and $\mathbb{E}[\nu(\Delta_p)]$ in closed form. To decide *which* forecast to acquire, we compare them; since the baseline term $(\mu_p - (n+1)\mu_r)^+$ is common to (3.7)–(3.8), it cancels

$$\mathbb{E}[\nu(\Delta_p)] - \mathbb{E}[\nu(\Delta_r)] = \mathbb{E}[(\Delta_p - (n+1)\mu_r)^+] - \mathbb{E}[(\mu_p - (n+1)\Delta_r)^+]. \tag{4.1}$$

We assume the following priors, independent as stipulated in Section 3.4. The release delay Δ_r has a density f_r on $[0,\infty)$; the resequencing gain Δ_p has distribution F_p, survival function $\overline{F}_p = 1 - F_p$, and mean μ_p and may be negative, zero, or positive. We write σ_r^2 and σ_p^2 for the two variances *when they exist* and use them only in the qualitative discussion of Section 4.3: No result below requires either to be finite, and the heaviest gain law considered in Section 5.5, a power law of index $\alpha = 1.5$, has $\sigma_p^2 = \infty$. We evaluate (4.1) asymptotically in the queue size n.

4.2. Asymptotic comparison

Proposition 4.1. *Assume $\mu_p > 0$, f_r is continuous at 0 with $f_r(0) > 0$, and $\mathbb{E}[(\Delta_p)^+] < \infty$. As $n \to \infty$ (so that $(n+1)\mu_r > \mu_p$ and the baseline term vanishes),*

$$\mathbb{E}[\nu(\Delta_r)] = \frac{f_r(0)\,\mu_p^2}{2(n+1)} + o\Big(\tfrac{1}{n}\Big) = \Theta\Big(\tfrac{1}{n+1}\Big), \qquad \mathbb{E}[\nu(\Delta_p)] = \int_{(n+1)\mu_r}^{\infty} \overline{F}_p(y)\,dy.$$

Consequently,

$$\frac{\mathbb{E}[\nu(\Delta_p)]}{\mathbb{E}[\nu(\Delta_r)]} \to \infty \quad \Longleftrightarrow \quad \int_{(n+1)\mu_r}^{\infty} \overline{F}_p(y)\,dy = \omega\Big(\tfrac{1}{n}\Big).$$

Proof. For n large enough that $(n+1)\mu_r > \mu_p$, the baseline term $(\mu_p - (n+1)\mu_r)^+$ vanishes in both (3.7) and (3.8). Writing $m = n+1$ and substituting $u = mx$,

$$\mathbb{E}[(\mu_p - m\Delta_r)^+] = \int_0^{\mu_p/m} (\mu_p - mx)\, f_r(x)\,dx = \frac{1}{m}\int_0^{\mu_p} (\mu_p - u)\, f_r(u/m)\,du.$$

The upper limit μ_p/m is positive precisely because $\mu_p > 0$; were $\mu_p \le 0$, the integrand would vanish identically, and $\mathbb{E}[\nu(\Delta_r)]$ would be exactly 0 rather than of order $1/m$, which is why that assumption is needed. By continuity of f_r at 0, $f_r(u/m) \to f_r(0)$, and $\int_0^{\mu_p}(\mu_p - u)\,du = \mu_p^2/2$, which gives the stated $\Theta(1/m)$ rate. For $\mathbb{E}[\nu(\Delta_p)]$, the identity $\mathbb{E}[(X-K)^+] = \int_K^{\infty} \overline{F}_X(y)\,dy$ with $K = m\mu_r$ yields the integral. Comparing the two orders of decay gives the equivalence. □

Corollary 4.2 (Decay of release-delay information)**.** *Under the assumptions of Proposition 4.1, $\mathbb{E}[\nu(\Delta_r)] = \Theta(1/(n+1)) \to 0$: The value of release-delay information vanishes inversely with congestion. This is immediate from the first display of Proposition 4.1, whose leading term is positive because $f_r(0) > 0$.*

Remark 4.1. The comparison is therefore governed by the *upper tail of* Δ_p, not by congestion alone. If Δ_p has a heavy (power-law) tail $\overline{F}_p(y) \sim c\,y^{-\alpha}$ with $1 < \alpha < 2$, then $\int_{m\mu_r}^{\infty} \overline{F}_p(y)\,dy \sim c'\,m^{1-\alpha} = \omega(1/m)$, processing-gap information dominates, and the ratio diverges. Conversely, if the gap has all moments

finite (e.g., the lognormal used in Section 5, which despite its pronounced skew is lighter than any power law), $\mathbb{E}[\nu(\Delta_p)]$ decays faster than $1/n$, and release-delay information eventually dominates. The practically relevant range is moderate n: For workloads whose job sizes follow a power law with $\alpha < 2$, as documented for process lifetimes and file transfers in computing systems [22, 23], Δ_p information is the more valuable acquisition throughout the operating range (Section 5), the asymptotic crossover occurring only at queue sizes far beyond those of interest.

4.2.1. Joint information acquisition

When both Δ_r and Δ_p are observed, the agent can determine whether $\Delta_p > (n+1)\Delta_r$ exactly, making the optimal decision (wait or not) deterministic. The value of joint information is

$$\mathbb{E}[\nu(\Delta_r, \Delta_p)] = \mathbb{E}\big[(\Delta_p - (n+1)\Delta_r)^+\big] \; - \; (\mu_p - (n+1)\mu_r)^+. \tag{4.2}$$

By monotonicity of information, joint information cannot be worse than either single source:

$$\mathbb{E}[\nu(\Delta_r, \Delta_p)] \geq \max\left(\mathbb{E}[\nu(\Delta_r)],\, \mathbb{E}[\nu(\Delta_p)]\right). \tag{4.3}$$

In general, strict inequality may occur, i.e., joint acquisition can provide complementary value relative to one-parameter acquisition.

4.3. Decision criteria and strategic implications

The analytical results suggest a strategic framework for information acquisition. The choice between acquiring processing time gap information (Δ_p) or release time delay information (Δ_r) depends on the operational characteristics of the scheduling environment.

We identify three distinct regimes characterizing the VOI. First, in the *congested regime* ($n \gg 1$), the cost of waiting scales linearly with n and both information values decline. By Proposition 4.1, $\mathbb{E}[\nu(\Delta_r)]$ decays as $\Theta(1/n)$, so which source dominates is governed by the tail of Δ_p: A power-law gap (index $\alpha < 2$) keeps processing-gap information dominant by surfacing the rare opportunities ($\Delta_p \gg 0$) that justify the disruption, whereas a gap with all moments finite lets release-delay information overtake it at large n. Second, in the *high-variance regime*, where job sizes vary by orders of magnitude (e.g., the heavy-tailed job-size distributions documented in computing systems [22, 23]), Δ_p information is dominant because the potential gain can be arbitrarily large. Finally, in the *synchronized arrival regime*, where arrivals are tightly clustered (low σ_r) but uncertain, Δ_r information is critical to avoid suboptimal timing, such as failing to wait when beneficial or waiting unproductively.

We stress that the dominance of Δ_p in the congested regime is contingent on a sufficiently heavy upper tail of the gap distribution, as made precise by Proposition 4.1 and the following remark; the qualitative guidance in Table 2 should be read with this proviso in mind.

Combined scenarios: When a system exhibits both high congestion ($n \gg 1$) and high arrival uncertainty (σ_r large), the ranking again turns on the gap tail. For a power-law gap with $\alpha < 2$, Δ_p information remains dominant: Congestion amplifies the cost factor ($n+1$), and the surviving value comes entirely from the rare large gains, so precise release timing is of secondary importance. For a bounded or light-tailed gap, the opposite holds beyond the crossover of Table 3 and Δ_r should be acquired instead. The same dichotomy applies when arrivals are tightly synchronized (low σ_r), with the crossover simply displaced towards larger n, since a smaller σ_r lowers $\mathbb{E}[\nu(\Delta_r)]$ without affecting $\mathbb{E}[\nu(\Delta_p)]$.

Table 2. Strategic guide for information acquisition. The guidance is valid *under the stated condition on the upper tail of the gap distribution*, which Proposition 4.1 shows to be the governing factor; congestion alone does not determine the ranking.

System Characteristic	Tail of Δ_p	Preferred Info.	Rationale
Congested System ($n \gg 1$)	Power law, $\alpha < 2$	Processing Gap (Δ_p)	Waiting costs scale with n, but the gap tail decays more slowly than $\Theta(1/n)$, so rare large-gain opportunities keep justifying the disruption.
Congested System ($n \gg 1$)	Bounded, or all moments finite	Release Delay (Δ_r)	$\mathbb{E}[\nu(\Delta_p)]$ then decays faster than the $\Theta(1/n)$ decay of $\mathbb{E}[\nu(\Delta_r)]$, which therefore dominates beyond a finite crossover (Table 3).
High Job Variance (High σ_p^2)	Any	Processing Gap (Δ_p)	Greater dispersion of Δ_p implies more frequent outlier opportunities where waiting yields significant gains.
High Arrival Uncertainty (High σ_r^2)	Any	Release Delay (Δ_r)	High variability in Δ_r makes the default decision risky; information identifies rare low-delay instances.
Light System ($n \approx 0$)	Any	Release Delay (Δ_r)	Waiting costs are minimal; precise timing (Δ_r) optimizes the trade-off without penalty.

4.4. Optimal information acquisition strategy

When acquiring information is costly (c_r for observing Δ_r, c_p for Δ_p), the choice reduces to a plain cost-benefit comparison: Acquire whichever subset of the two forecasts maximizes the net value, that is, the expected value of the subset minus its acquisition cost. Writing $\mathcal{I} \subseteq \{\Delta_r, \Delta_p\}$ for the subset acquired, $c(\mathcal{I}) = \sum_{i \in \mathcal{I}} c_i$ for its cost, and adopting the conventions $\mathbb{E}[\nu(\varnothing)] = 0$ and $c(\varnothing) = 0$, the optimal acquisition is

$$\mathcal{I}^\star \in \arg\max_{\mathcal{I} \subseteq \{\Delta_r, \Delta_p\}} \Big\{ \mathbb{E}[\nu(\mathcal{I})] - c(\mathcal{I}) \Big\},$$

a comparison of four quantities, where $\mathbb{E}[\nu(\Delta_r, \Delta_p)]$ is the value of complete information given by (4.2). Acquiring nothing is optimal exactly when no subset achieves a positive net value.

5. Numerical experiments

5.1. Experimental scope

The experiments below evaluate the single-epoch VOI expressions (3.7)–(3.8) at a fixed queue size n, over the parameter ranges of interest. They quantify how $\mathbb{E}[\nu(\Delta_p)]$ and $\mathbb{E}[\nu(\Delta_r)]$ respond to congestion, to the dispersion of each parameter, and to the tail of the gain distribution, and they test numerically the prediction of Proposition 4.1, which is established only asymptotically. No horizon is simulated, and no queue evolves over time, so the reported magnitudes are the marginal value of one forecast at one decision epoch rather than the flow-time improvement attainable by a complete scheduler. Evaluation over full problem instances, with dynamic arrivals, a time-varying queue, and cumulative VOI across sequential now-or-later decisions, is left to future work (Section 5.7).

5.2. Experimental setup

Evaluation method The two quantities to be evaluated are the expectations appearing in (3.7)–(3.8). We compute both exactly rather than by simulation. The gap term $\mathbb{E}[(\Delta_p - (n+1)\mu_r)^+]$ admits a closed form for each law considered, and the delay term $\mathbb{E}[(\mu_p - (n+1)\Delta_r)^+]$ is obtained by quadrature against the truncated-normal density, its integrand being supported on the short interval $[0, \mu_p/(n+1)]$. Exactness matters because the comparison of Section 5.4 takes place in the far tail, where sampling error is largest: A 200,000-scenario Monte-Carlo estimate underestimates $\mathbb{E}[(\Delta_p - (n+1)\mu_r)^+]$ by 21% at $n = 100$ and 38% at $n = 200$ for the lognormal gap, precisely where the asymptotic crossover of Proposition 4.1 lies. The argument is stronger still for the Pareto gain with $\alpha = 1.5$ of Section 5.5, on which the paper's main comparative claim rests: that law has infinite variance, so the sample mean of $(\Delta_p - K)^+$ has no finite-variance central limit and the Monte-Carlo estimate does not settle at any practical sample size. Simulation is therefore not merely imprecise but unusable in exactly the regime of interest, whereas the closed form is exact. All figures and tables of this section are produced by a single self-contained script (`voi_simulation.py`), which lists the closed forms used.

Monte-Carlo validation of the closed forms As an independent check that the closed forms (3.7)–(3.8) correctly describe the decision problem, we also simulate the three policies directly over $N = 200{,}000$ independent scenarios, drawing Δ_p and Δ_r from their priors and applying the decision rules below. Over the range where the Monte-Carlo estimator is reliable, the two agree closely (relative deviation of 0.05% at $n = 5$ and below 4% for $n \le 20$), which validates the derivations; the deviation grows to 21% at $n = 100$ and 38% at $n = 200$ for the sampling reason just given. The comparison is reported in `exp0_mc_check.csv`.

Strategies and decision logic Three decision rules enter the VOI computation, each acting on a realization (Δ_p, Δ_r) of the parameters:

1. **No information (baseline)** – the reference policy a_{base}: relies solely on prior expectations, waiting if and only if $\mathbb{E}[\Delta_p] > (n+1)\mathbb{E}[\Delta_r]$.
2. Δ_p **Information** – the information policy $a_{\mathcal{I}}$ for $\mathcal{I} = \Delta_p$: observes the realization Δ_p but relies on the expectation $\mathbb{E}[\Delta_r]$, waiting if $\Delta_p > (n+1)\mathbb{E}[\Delta_r]$.
3. Δ_r **Information** – the information policy $a_{\mathcal{I}}$ for $\mathcal{I} = \Delta_r$: observes Δ_r but relies on $\mathbb{E}[\Delta_p]$, waiting if $\mathbb{E}[\Delta_p] > (n+1)\Delta_r$.

All curves in this section report the expected cost reduction of rules 2 and 3 relative to rule 1, which is exactly (3.7)–(3.8).

Baseline The no-information baseline $a_{\text{base}} = a_{\varnothing}$ is exactly the expectation rule of Section 3.4 (wait if and only if $\mathbb{E}[\Delta_p] > (n+1)\mathbb{E}[\Delta_r]$). The three rules above therefore define precisely the model underlying (3.7)–(3.8), and every curve reported below is an evaluation of those formulas rather than of a separate cost model or of a full scheduling simulation.

Parameter settings Unless otherwise specified in the sensitivity analyses, the prior distributions are set as follows. We normalize the mean processing time $p_0 = 1$. The processing time gap Δ_p follows

a lognormal distribution with log-scale location 0 and shape parameter $\sigma_p = 1.0$, so that its mean is $\mu_p = \mathbb{E}[\Delta_p] = e^{1/2} \approx 1.65$; we reserve μ_p for this mean throughout, consistently with Sections 3.4–4, generating a strongly skewed distribution typical of computing workloads. The release time delay Δ_r follows a truncated normal distribution with mean $\mu_r = 0.2$ and pre-truncation scale $\sigma_r = 0.1$, representing moderate arrival variability.

A note on σ_p and σ_r in this section In Section 4, σ_p^2 and σ_r^2 denote the variances of Δ_p and Δ_r. In the present section, the same symbols denote the *shape* parameters of the two chosen families: σ_p is the shape of the lognormal and σ_r the scale of the normal before truncation. Neither equals the corresponding standard deviation. A lognormal of shape $\sigma_p = 1$ has standard deviation 2.16, and one of shape 2.5 has standard deviation 517; the truncated normal of scale $\sigma_r = 0.1$ has standard deviation 0.094. Both parameters are, however, strictly increasing measures of dispersion over the ranges swept in the sensitivity analysis below, which is all they are used for. In the variance sweeps, we fix the queue size at $n = 10$ and vary one parameter at a time; in the σ_r sweep, the truncated normal is re-centered so that $\mu_r = 0.2$ stays fixed, isolating a mean-preserving spread.

Value of information Writing $L(\cdot)$ for the cost of an action relative to dispatching now (Section 3.4), the value of an information source $\mathcal{I}$ is the expected cost reduction $\mathbb{E}\big[L(a_{\text{base}}) - L(a_{\mathcal{I}})\big]$, which is exactly (3.7) for $\mathcal{I} = \Delta_r$ and (3.8) for $\mathcal{I} = \Delta_p$. All curves in this section report this quantity, evaluated as described above. The corresponding Monte-Carlo estimator, used for the validation check only, is $\frac{1}{N}\sum_{k=1}^{N}\big[L(a_{\text{base}}^{(k)}) - L(a_{\mathcal{I}}^{(k)})\big]$ over the N sampled scenarios.

5.3. Impact of queue size (n)

Figure 1 illustrates the marginal value of knowing Δ_p versus Δ_r as the queue size n increases from 1 to 49.

Both values rise for very small queues, peak at single-digit n, and then decline as congestion makes waiting rarely beneficial. Over the entire practical range, processing-gap information remains the more valuable acquisition. Consistently with Corollary 4.2, $\mathbb{E}[\nu(\Delta_r)]$ decays as $\Theta(1/n)$; because the lognormal gap is light relative to a power law, $\mathbb{E}[\nu(\Delta_p)]$ decays faster still, so the two curves eventually cross; this regime is examined next.

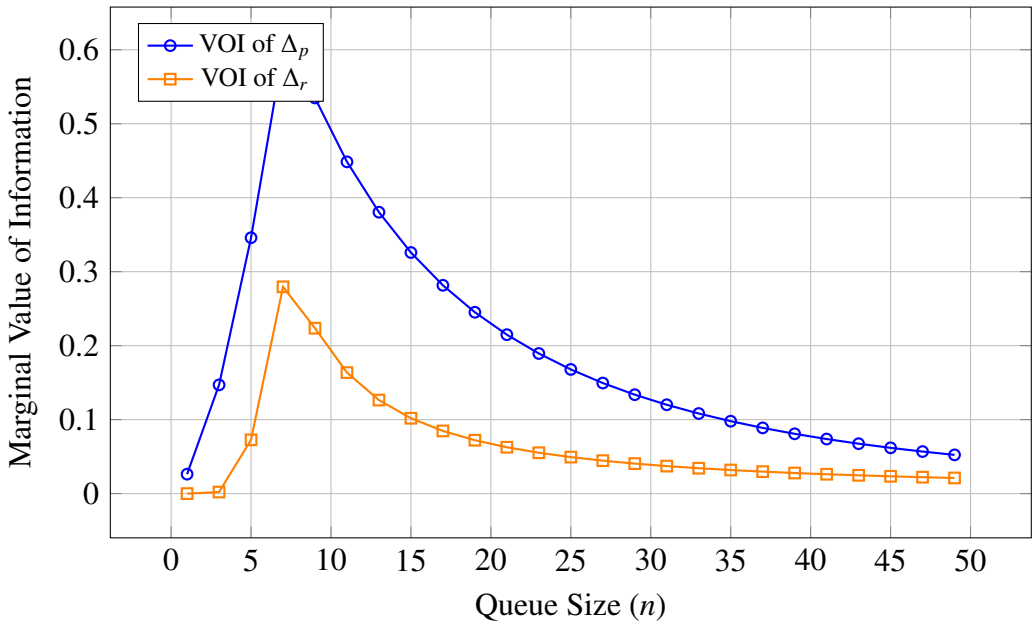

Figure 1. VOI versus queue size n over the practical range ($n \leq 49$) for a lognormal gap Δ_p ($\sigma_p = 1$) and a truncated-normal delay Δ_r. Processing-gap information (Δ_p) dominates release-delay information (Δ_r) throughout this range; both peak at small n and decline as congestion makes waiting rarely beneficial. The asymptotic crossover to Δ_r-dominance occurs only at much larger n (Figure 2), and its location depends on the tail of the gap distribution (Figure 3). The curves evaluate the closed-form expressions (3.7)–(3.8) at a single decision epoch; they are not the output of a dynamic scheduling simulation.

5.4. Asymptotic crossover: direct validation of Proposition 4.1

To test the asymptotic prediction beyond the range of Figure 1, we evaluate the closed-form values (3.7)–(3.8) exactly (lognormal gap Δ_p and truncated-normal delay Δ_r as above) for queue sizes up to $n = 1000$. Figure 2 reports both values on a log–log scale.

The result confirms Proposition 4.1: $\mathbb{E}[v(\Delta_r)]$ follows the predicted $\Theta(1/n)$ decay, while $\mathbb{E}[v(\Delta_p)]$ falls off markedly faster. The curves cross at $n = 106$; below the crossover, processing-gap information is the more valuable acquisition, and above it release-delay information dominates. This is precisely the tail-dependent behavior of the corrected theory: A heavy-tailed (power-law, index $\alpha < 2$) gap pushes the crossover to arbitrarily large n and recovers the unconditional dominance of Δ_p, whereas a light tail makes the dominance a moderate-n phenomenon. Section 5.5 verifies both halves of this statement directly by varying the gap law. The practically relevant operating range (small-to-moderate n) therefore favors Δ_p information, consistent with Figure 1.

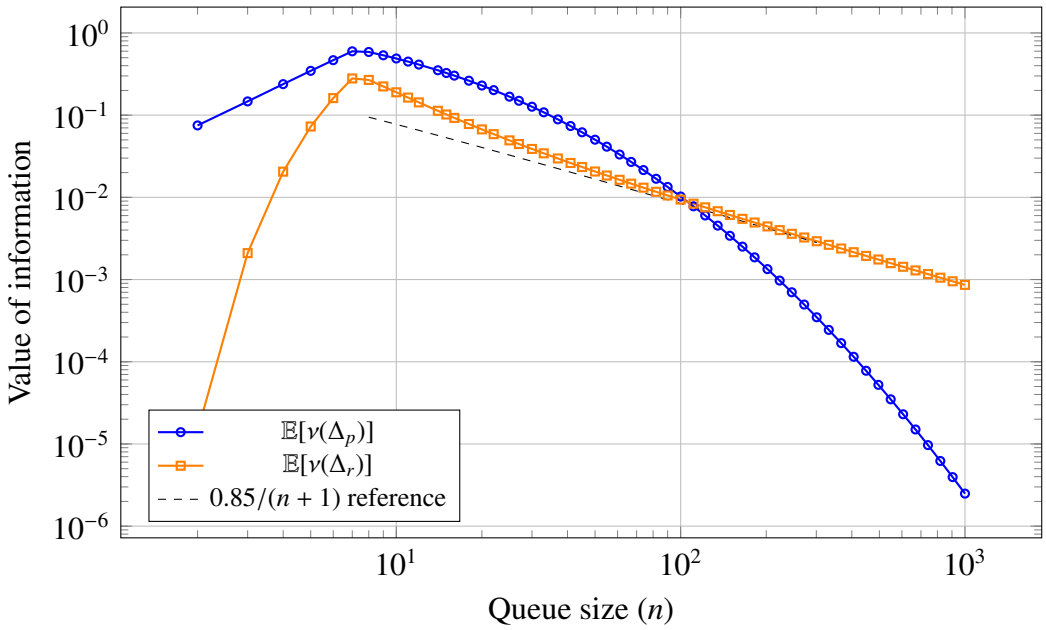

Figure 2. Values up to $n = 1000$ (log–log scale), lognormal gap Δ_p. The release-delay value $\mathbb{E}[v(\Delta_r)]$ tracks the leading term of Proposition 4.1 (dashed, $f_r(0)\mu_p^2/[2(n+1)]$ with $f_r(0)\mu_p^2/2 \approx 0.85$), whereas the gap value $\mathbb{E}[v(\Delta_p)]$ decays faster because the lognormal gap is light relative to a power law. The two cross near $n \approx 106$. The curves evaluate the closed-form expressions (3.7)–(3.8) at a single decision epoch; they are not the output of a dynamic scheduling simulation.

5.5. *Sensitivity to the tail of the gap distribution*

Proposition 4.1 makes the comparison between the two information sources depend on the upper tail of Δ_p and not on congestion alone. Figures 1 and 2 exhibit a single gap law, so they cannot separate the two effects. This subsection does so directly by evaluating (3.7)–(3.8) for five gap families spanning the range of tail behaviors, all calibrated to the *same* mean $\mu_p = e^{1/2} \approx 1.65$:

- **Uniform** on $[0, 2\mu_p]$: bounded support, the case in which Δ_p cannot exceed a fixed ceiling;
- **Exponential** with mean μ_p: unbounded but light-tailed, $\overline{F}_p$ decaying geometrically;
- **Lognormal**$(0, 1)$: the baseline of Sections 5.3–5.4, strongly skewed with all moments finite;
- **Pareto** with $\alpha = 2.5$ and scale $x_m = \mu_p(\alpha - 1)/\alpha$: a genuine power law with *finite* variance;
- **Pareto** with $\alpha = 1.5$: a genuine power law with *infinite* variance, i.e., the regime $1 < \alpha < 2$ of Proposition 4.1.

Fixing the mean is what makes the comparison meaningful: The baseline policy $a_\varnothing$, the threshold $(n+1)\mu_r$, and the whole curve $\mathbb{E}[v(\Delta_r)]$ are then identical across the five families, so any difference between them is attributable to the tail alone. Figure 3 reports the five gap curves against the single common release-delay curve.

The picture matches the theory point by point. The crossover queue sizes, defined as the smallest n from which release-delay information dominates, are reported in Table 3.

Four observations deserve comment.

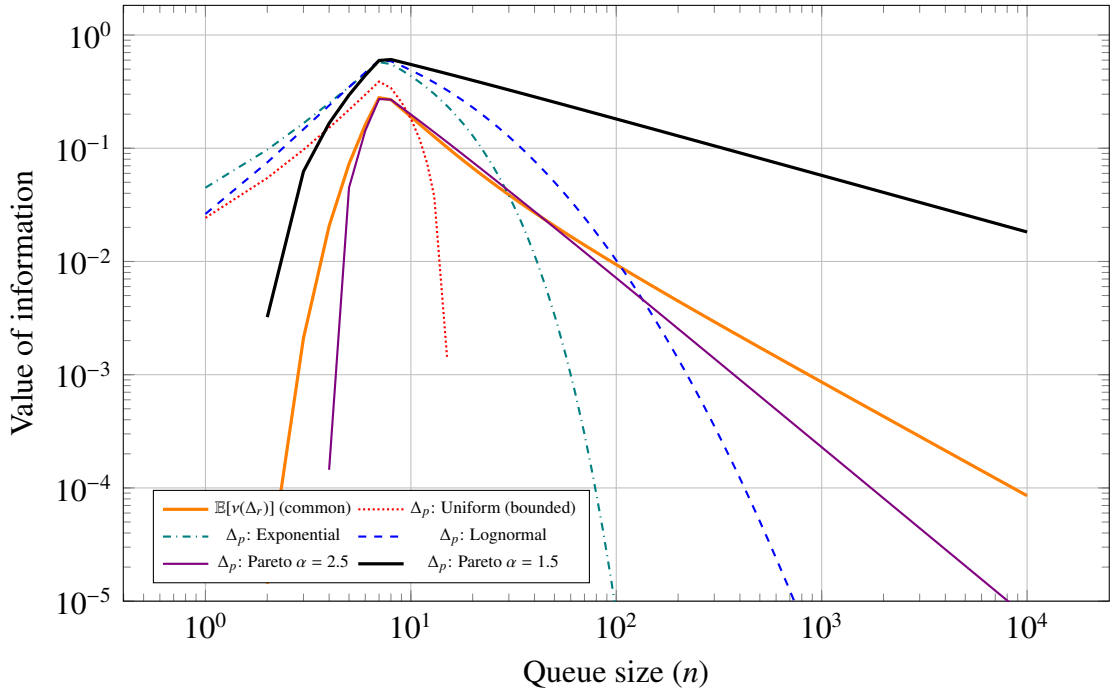

Figure 3. Value of processing-gap information for five gap distributions with a common mean $\mu_p \approx 1.65$, against the common release-delay curve (orange), log–log scale, $n \leq 10^4$. The heavier the tail of Δ_p, the later $\mathbb{E}[\nu(\Delta_p)]$ falls below $\mathbb{E}[\nu(\Delta_r)]$; for the power law with $\alpha = 1.5 < 2$ it never does, as Proposition 4.1 predicts. The curves evaluate the closed-form expressions (3.7)–(3.8) at a single decision epoch; they are not the output of a dynamic scheduling simulation.

Table 3. Crossover queue size $n^\star$ beyond which release-delay information dominates by gain distribution (common mean $\mu_p \approx 1.65$, $\mu_r = 0.2$, $\sigma_r = 0.1$). A power law with $\alpha < 2$ removes the crossover entirely. Rows are ordered by $n^\star$, which is not a monotone function of tail weight: The heavier-tailed Pareto $\alpha = 2.5$ crosses before the lognormal for the reason given in the text.

Gap distribution	Tail behavior	Variance	Crossover $n^\star$
Uniform $[0, 2\mu_p]$	bounded support	finite	10
Exponential	light (geometric)	finite	30
Pareto $\alpha = 2.5$	power law	finite	44
Lognormal $\sigma_p = 1$	all moments finite	finite	106
Pareto $\alpha = 1.5$	power law, $\alpha < 2$	infinite	none ($> 10^4$)

First, the qualitative dichotomy of Proposition 4.1 is reproduced exactly: The crossover exists for all four gap laws whose tail integral decays faster than $\Theta(1/n)$, and only for those. For the bounded gap, $\mathbb{E}[v(\Delta_p)]$ reaches zero at a *finite* queue size ($n = 16$, where $(n+1)\mu_r$ exceeds the support upper bound $2\mu_p$), so gap information becomes strictly worthless while release-delay information is still positive; this is the most extreme form of the reversal.

Second, the value of $n^\star$ is *not* a monotone function of tail weight, and the comparison of the lognormal ($n^\star = 106$) with the strictly heavier-tailed Pareto $\alpha = 2.5$ ($n^\star = 44$) shows why. Proposition 4.1 constrains the asymptotic decay rate, not the position of the curve at moderate n, and the two orderings need not agree: With a common mean, the Pareto $\alpha = 2.5$ law concentrates more mass near its scale $x_m = 0.99$ and less in the intermediate range than the lognormal, so its value curve lies *below* the lognormal one until $n = 144$, and it therefore meets the release-delay curve earlier despite decaying more slowly thereafter. The asymptotic ordering does assert itself; beyond $n = 144$, the Pareto $\alpha = 2.5$ value exceeds the lognormal value, and the gap widens without bound. Practitioners should read Table 3 accordingly, as a set of computed thresholds for the specific laws considered rather than as a ranking that a tail index alone would predict.

Third, the case $\alpha = 2.5$ also shows that being a power law is not by itself sufficient: With $\alpha > 2$, the integral $\int_{(n+1)\mu_r}^{\infty} \overline{F}_p = \Theta(n^{1-\alpha})$ decays faster than $\Theta(1/n)$, and the crossover occurs at a finite $n^\star = 44$. It is the index condition $\alpha < 2$, not the power-law form, that drives the asymptotic dominance of Δ_p.

Fourth, for $\alpha = 1.5$, the prediction is quantitative and can be checked as such. Proposition 4.1 gives $\mathbb{E}[v(\Delta_p)]/\mathbb{E}[v(\Delta_r)] = \Theta(n^{2-\alpha}) = \Theta(n^{1/2})$, so the ratio should grow by a factor $\sqrt{10} \approx 3.16$ per decade of n. The computed ratios are 18.7 at $n \approx 10^2$, 67.8 at $n \approx 10^3$, and 213.4 at $n = 10^4$, i.e., successive factors of 3.62 and 3.15, converging to the predicted rate.

Practically, these results sharpen the managerial guidance of Section 4. The workloads for which Δ_p forecasting remains the right investment at high congestion are those with genuinely heavy-tailed resequencing gains, the regime documented for computing workloads [22, 23]; where gaps are bounded by design (fixed batch sizes, standardized operations), congestion favors arrival-time forecasting instead and does so at surprisingly small queue sizes.

5.6. Impact of heterogeneity and uncertainty

We further analyzed how the variance of the parameters affects information value (Figure 4).

These sensitivities confirm the asymmetry of (3.7)–(3.8). Greater gap heterogeneity (σ_p, left panel) sharply raises the value of observing Δ_p, which exploits the upper tail of the gap; once gaps are very large, waiting is almost always optimal, and release-delay information loses all value. Symmetrically, greater (mean-preserving) arrival uncertainty (σ_r, right panel) raises the value of observing Δ_r, while leaving the value of Δ_p exactly unchanged, since (3.8) involves Δ_r only through its mean, which the sweep holds fixed.

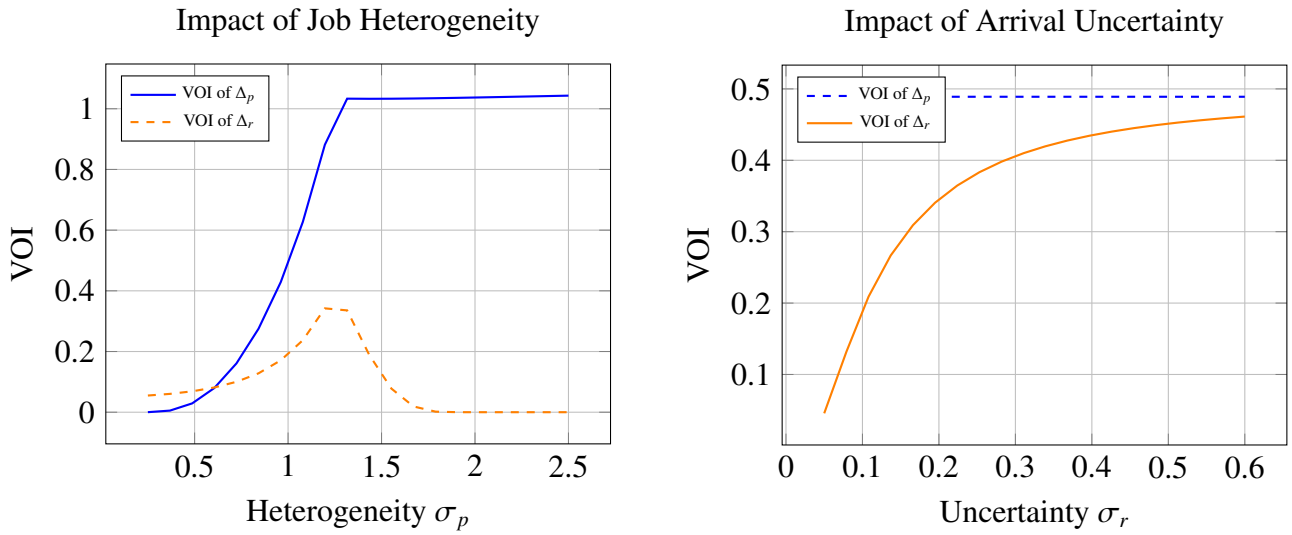

Figure 4. Sensitivity at fixed queue size $n = 10$, lognormal gap Δ_p, and truncated-normal delay Δ_r. Left: Increasing gap heterogeneity σ_p sharply raises $\mathbb{E}[v(\Delta_p)]$, while $\mathbb{E}[v(\Delta_r)]$ falls toward zero (with very large gaps, one should almost always wait, so the delay becomes uninformative). Right: Increasing arrival uncertainty σ_r (mean-preserving) raises $\mathbb{E}[v(\Delta_r)]$, while $\mathbb{E}[v(\Delta_p)]$ is exactly constant, since it depends on Δ_r only through its fixed mean. The curves evaluate the closed-form expressions (3.7)–(3.8) at a single decision epoch; they are not the output of a dynamic scheduling simulation.

5.7. Scope and limitations

We acknowledge several limitations of this study. First, the analytical core is a single local decision epoch involving one announced future job, not an online scheduling policy. We derive no competitive ratio, and none of our results should be read as a performance guarantee for a complete algorithm; the value formulas quantify the marginal worth of one forecast at one decision point. Formal extension to sequential, full-horizon settings (dynamic arrivals, a time-varying queue, and cumulative VOI, where successive decisions interact through the evolving queue) is the main direction for future work. Second, Corollary 3.6 (one bit suffices) holds for the isolated binary problem; it does not assert that one bit per epoch drives an optimal full-horizon policy, since different bits might be optimal at different stages of a full horizon, and the relevant threshold itself shifts with the queue state. Third, the *asymptotic* dominance of Δ_p requires a genuinely heavy (power-law, index $\alpha < 2$) gap distribution; for the lognormal used here or for bounded gaps, release-delay information prevails at large n (Proposition 4.1, Table 3), and Δ_p dominance holds only over the moderate-n operating range. Fourth, the numerical section evaluates the closed-form value expressions over a prior; it does not validate the framework against a scheduling workload, and empirical estimation of the gap and delay distributions from operational data remains to be done. Finally, our results are specific to the $\sum C_j$ objective; makespan or tardiness objectives would yield different value functions.

6. Managerial insights and practical applications

The integration of machine learning and data-driven approaches in manufacturing [24] makes real-time scheduling decisions increasingly viable but requires quantifiable metrics for information value to justify investments in sensing and forecasting infrastructure. Our framework directly addresses this need by providing explicit formulas for computing the marginal value of different types of information. The discussion below transposes the single-epoch results to three settings in which now-or-later choices recur; it is offered as guidance on where to direct forecasting effort, not as a claim that the model captures those systems in full.

Consider electric vehicle charging infrastructure [25, 26], where operators must decide whether to invest in sensors that predict arrival times versus those that forecast battery capacities. Our analysis shows that battery size forecasting (measuring Δ_p, the gap between current and incoming vehicle charge requirements) is the more valuable acquisition in congested stations when charge requirements are strongly heterogeneous, since only large capacity gaps then justify the idle time imposed on the waiting vehicles; for homogeneous demand, congestion instead erodes this advantage (Proposition 4.1). Table 3 makes the threshold concrete: If charge requirements are effectively bounded, as with a homogeneous fleet, arrival-time forecasting becomes the better investment from about 10 vehicles queued onwards, whereas a mixed fleet with occasional very large deficits keeps capacity forecasting ahead indefinitely. In contrast, arrival time prediction (measuring Δ_r) is more valuable during off-peak hours when stations have spare capacity and precise timing becomes the differentiator.

In operating room scheduling [27] and more broadly in online stochastic health-care scheduling [28], hospitals face similar trade-offs when deciding whether to delay elective surgeries to accommodate emergency cases. Knowing the processing time gap Δ_p (difference between the scheduled surgery duration and the incoming emergency duration) informs these decisions. This formalization transforms informal clinical judgment into data-driven protocols.

Smart manufacturing environments present another application. In flexible production lines handling low-volume, high-mix orders, knowing whether $\Delta_p > (n+1)\Delta_r$, that is, whether waiting is beneficial, amounts to a single bit of binary advice at each dispatching decision [4]. Corollary 3.6 establishes this for one such decision in isolation; whether a per-decision bit remains sufficient once the decisions are chained over a horizon is an open question, and we do not claim it here. Multi-agent reinforcement learning systems [29] demonstrate that distributed AI schedulers can exploit such real-time sensor data to coordinate complex production sequences. Our framework formalizes when these forecasting investments are justified by quantifying the marginal value of additional information.

7. Conclusion and future perspectives

This paper introduced a preference-based framework to quantify information value in online scheduling, connecting worst-case competitive analysis and practical semi-online decision-making. By characterizing ignorance as a set of Pareto optimal actions, we showed that valuable information is that which eliminates dominated strategies.

We applied the framework to one specific building block of the single machine problem $1|online, r_j|\sum C_j$: the local, binary now-or-later decision at a single epoch with one announced arrival. The restriction is what makes the analysis exact, and it also bounds what the results claim. We do not

propose an online scheduling policy, and we prove no competitive ratio; what we obtain is the marginal value of a forecast at one decision point, with the understanding that a full policy chains such decisions in a way our model does not yet capture. Within this scope, our analysis revealed three insights. First, the VOI is highly context-dependent: The ranking of the two sources is governed by the upper tail of the gap distribution rather than by congestion alone. Processing gap information (Δ_p) dominates at every queue size when the gap follows a power law of index $\alpha < 2$, whereas for bounded or light-tailed gaps, release-delay information (Δ_r) takes over beyond a finite crossover, which our numerical evaluation places between $n = 10$ and $n = 106$ depending on the law. Δ_r also remains the more valuable source in synchronized, low-load systems regardless of the tail. Second, precise values are often unnecessary; for this binary decision, the optimal action depends on the two parameters only through the single comparison $\Delta_p > (n + 1)\Delta_r$ between the resequencing gain and the total waiting cost imposed on the $n + 1$ affected jobs, so one bit of advice is necessary and sufficient, and exact parameter values add nothing. Third, the derived expected value formulas allow decision-makers to optimize forecasting investments strategically based on system parameters (queue size, volatility, and above all the tail of the gain distribution) rather than intuition.

The framework presented here suggests avenues for future research. A natural extension is to multi-machine systems, including parallel machines ($P|online, r_j|\sum C_j$) and flow shops ($F|online, r_j|\sum C_j$), for which online models have already been studied [30]. Preliminary analysis suggests that in such settings, processing time gap information on bottleneck machines would have the highest value, as these machines constrain the overall system performance. Another important direction is to apply the framework to alternative performance objectives beyond total completion time, such as makespan ($C_{\max}$), total tardiness ($\sum T_j$), and energy consumption, which are critical in modern manufacturing environments.

Furthermore, while the current work assumes a single decision point, many real-world scenarios involve dynamic information acquisition. Modeling such scenarios would require multi-stage stochastic programming formulations [31] to capture the option value of waiting for more information. Integration with machine learning represents another area for development; by using historical data to estimate distributions, practitioners could compute data-driven estimates of information value. Finally, empirical validation on real-world datasets, such as manufacturing job shop logs, healthcare surgery schedules, and electric vehicle charging arrival data, is essential to bridge theory and practice, validating theoretical predictions against practical challenges like non-stationary distributions and correlated arrivals.

Collectively, these directions contribute to more adaptive scheduling systems that effectively balance performance guarantees with the cost of information acquisition.

Author contributions

Maria Zemzami: Conceptualization, Methodology, Formal analysis, Investigation, Visualization, Writing (original draft).
Nhan-Quy Nguyen: Conceptualization, Methodology, Formal analysis, Software, Validation, Project administration, Writing (review and editing).
Farouk Yalaoui: Conceptualization, Supervision, Funding acquisition, Writing (review and editing).
Yassine Ouazene: Methodology, Validation, Supervision, Writing (review and editing).
All authors have read and approved the final version of the manuscript.

Use of Generative-AI tools declaration

The authors declare that generative artificial intelligence tools were used solely for language editing. These tools contributed neither to the scientific content, the derivations and proofs, the design and execution of the numerical experiments, nor to the interpretation of the results. The authors reviewed and edited the resulting text and take full responsibility for the content of the publication.

Acknowledgments

This research was supported by the Chaire Connected Innovation at the University of Technology of Troyes. The authors thank the anonymous reviewers for constructive feedback.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 5th edition, Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-26580-3
2. A. Borodin, R. El-Yaniv, *Online Computation and Competitive Analysis*, Cambridge University Press, Cambridge, 1998.
3. S. Albers, Better bounds for online scheduling, *SIAM J. Comput.*, **29** (1999), 459–473. https://doi.org/10.1137/S0097539797324874
4. Z. Tan, A. Zhang, Online and semi-online scheduling, in *Handbook of Combinatorial Optimization*, 2nd edition, Springer, New York, 2013, 2191–2252. https://doi.org/10.1007/978-1-4419-7997-1_2
5. L. Epstein, A survey on makespan minimization in semi-online environments, *J. Sched.*, **21** (2018), 269–284. https://doi.org/10.1007/s10951-018-0567-z
6. R. L. Graham, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, *Ann. Discrete Math.*, **5** (1979), 287–326. https://doi.org/10.1016/S0167-5060(08)70356-X
7. U. Schwiegelshohn, Job scheduling, in *Introduction to Scheduling* (eds. Y. Robert and F. Vivien), Chapman & Hall/CRC, Boca Raton, 2009, 79–102. https://doi.org/10.1201/9781420072747-c4
8. J. R. Correa, M. R. Wagner, LP-based online scheduling: from single to parallel machines, *Math. Program.*, **119** (2009), 109–136. https://doi.org/10.1007/s10107-007-0204-7
9. J. Carlier, E. Pinson, An algorithm for solving the job-shop problem, *Manag. Sci.*, **35** (1989), 164–176. https://doi.org/10.1287/mnsc.35.2.164
10. S. Albers, On randomized online scheduling, in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, ACM, New York, 2002, 134–143. https://doi.org/10.1145/509907.509930

11. C. A. Phillips, C. Stein, E. Torng, J. Wein, Optimal time-critical scheduling via resource augmentation, *Algorithmica*, **32** (2002), 163–200. https://doi.org/10.1007/s00453-001-0068-9
12. C. Chekuri, S. Im, B. Moseley, Minimizing maximum response time and delay factor in broadcast scheduling, in *Algorithms - ESA 2009*, Lecture Notes in Comput. Sci., 5757, Springer, Berlin, 2009, 444–455. https://doi.org/10.1007/978-3-642-04128-0_40
13. S. Im, B. Moseley, An online scalable algorithm for average flow time in broadcast scheduling, *ACM Trans. Algorithms*, **8** (2012), 39. https://doi.org/10.1145/2344422.2344429
14. D. Dwibedy, R. Mohanty, Semi-online scheduling: a survey, *Comput. Oper. Res.*, **139** (2022), 105646. https://doi.org/10.1016/j.cor.2021.105646
15. S. Dobrev, R. Královič, D. Pardubská, How much information about the future is needed?, in *SOFSEM 2008: Theory and Practice of Computer Science*, Lecture Notes in Comput. Sci., 4910, Springer, Berlin, 2008, 247–258. https://doi.org/10.1007/978-3-540-77566-9_21
16. H. J. Böckenhauer, D. Komm, R. Královič, R. Královič, T. Mömke, Online algorithms with advice: the tape model, *Inf. Comput.*, **254** (2017), 59–83. https://doi.org/10.1016/j.ic.2017.03.001
17. T. Lykouris, S. Vassilvitskii, Competitive caching with machine learned advice, *J. ACM*, **68** (2021), 1–25. https://doi.org/10.1145/3447579
18. M. Purohit, Z. Svitkina, R. Kumar, Improving online algorithms via ML predictions, in *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, Curran Associates, Red Hook, 2018, 9661–9670.
19. M. Mitzenmacher, S. Vassilvitskii, Algorithms with predictions, *Commun. ACM*, **65** (2022), 33–35. https://doi.org/10.1145/3528087
20. J. K. Lenstra, A. H. G. Rinnooy Kan, P. Brucker, Complexity of machine scheduling problems, *Ann. Discrete Math.*, **1** (1977), 343–362. https://doi.org/10.1016/S0167-5060(08)70743-X
21. W. E. Smith, Various optimizers for single-stage production, *Nav. Res. Logist. Q.*, **3** (1956), 59–66. https://doi.org/10.1002/nav.3800030106
22. M. Harchol-Balter, A. B. Downey, Exploiting process lifetime distributions for dynamic load balancing, *ACM Trans. Comput. Syst.*, **15** (1997), 253–285. https://doi.org/10.1145/263326.263344
23. M. E. Crovella, A. Bestavros, Self-similarity in World Wide Web traffic: evidence and possible causes, *IEEE/ACM Trans. Netw.*, **5** (1997), 835–846. https://doi.org/10.1109/90.650143
24. A. Göppert, L. Kaven, J. Baum, O. Melnychuk, R. Schmitt, Machine learning for online scheduling in manufacturing: a systematic literature review, *Procedia CIRP*, **130** (2024), 154–160. https://doi.org/10.1016/j.procir.2024.10.070
25. A. H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, A. Leon-Garcia, Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid, *IEEE Trans. Smart Grid*, **1** (2010), 320–331. https://doi.org/10.1109/TSG.2010.2089069
26. E. Sortomme, M. A. El-Sharkawi, Optimal scheduling of vehicle-to-grid energy and ancillary services, *IEEE Trans. Smart Grid*, **3** (2012), 351–359. https://doi.org/10.1109/TSG.2011.2164099
27. D. Duma, R. Aringhieri, An online optimization approach for the real time management of operating rooms, *Oper. Res. Health Care*, **7** (2015), 40–51. https://doi.org/10.1016/j.orhc.2015.08.006

28. A. Legrain, M. A. Fortin, N. Lahrichi, L. M. Rousseau, Online stochastic optimization of radiotherapy patient scheduling, *Health Care Manag. Sci.*, **18** (2015), 110–123. https://doi.org/10.1007/s10729-014-9270-6

29. T. Zhou, D. Tang, H. Zhu, Z. Zhang, Multi-agent reinforcement learning for online scheduling in smart factories, *Robot. Comput. Integr. Manuf.*, **72** (2021), 102202. https://doi.org/10.1016/j.rcim.2021.102202

30. K. Lee, F. Zheng, M. L. Pinedo, Online scheduling of ordered flow shops, *Eur. J. Oper. Res.*, **272** (2019), 50–60. https://doi.org/10.1016/j.ejor.2018.06.008

31. R. H. Möhring, F. J. Radermacher, G. Weiss, Stochastic scheduling problems I: general strategies, *Z. Oper. Res.*, **28** (1984), 193–260. https://doi.org/10.1007/BF01919323

AIMS Press