

## MIT Open Access Articles

### *Scheduling with Testing*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

**Citation:** Levi, Retsef et al. "Scheduling with Testing." *Management Science* 65, 2 (February 2019): v-vi, 459-954 © 2018 INFORMS

**As Published:** <http://dx.doi.org/10.1287/mnsc.2017.2973>

**Publisher:** Institute for Operations Research and the Management Sciences (INFORMS)

**Persistent URL:** <https://hdl.handle.net/1721.1/128492>

**Version:** Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

**Terms of use:** Creative Commons Attribution-Noncommercial-Share Alike



Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

# Scheduling with Testing

Retsef Levi

Sloan School of Management, Massachusetts Institute of Technology,  
77 Massachusetts Avenue, Cambridge, Massachusetts 02139,  
retsef@mit.edu

Thomas Magnanti

Institute Professor, Massachusetts Institute of Technology,  
77 Massachusetts Avenue, Cambridge, Massachusetts 02139,  
magnanti@mit.edu  
Singapore University of Technology and Design,  
20 Dover Drive, Singapore 138682,  
magnanti@sutd.edu.sg

Yaron Shaposhnik

Operations Research Center, Massachusetts Institute of Technology,  
77 Massachusetts Avenue, Cambridge, Massachusetts 02139,  
shap@mit.edu

We study a new class of scheduling problems that capture common settings in service environments, in which one has to serve a collection of jobs that have a-priori uncertain attributes (e.g., processing times and priorities) and the service provider has to decide how to dynamically allocate resources (e.g., people, equipment and time) between testing (diagnosing) jobs to learn more about their respective uncertain attributes and processing jobs. The former could inform future decisions, but could delay the service time for other jobs, while the latter directly advances the processing of the jobs but require making decisions under uncertainty. Through novel analysis we obtain surprising structural results of optimal policies that provide operational managerial insights, efficient optimal and near-optimal algorithms and quantification of the value of testing. We believe that our approach will lead to further research to explore this important practical tradeoff.

*Key words:* Scheduling, Dynamic Programming, Service Operations, Approximation Algorithms

---

## 1. Introduction

Effective management of many service systems often relies on the ability to appropriately classify and prioritize customers, tasks or jobs. However, in many settings the exact nature of the various jobs is uncertain; for example, the time and amount of resources required to process a given job and its relative priority might not be known exactly. While recent advancements in information

technologies enable obtaining far more accurate predictions about each job, there are still many settings, in which collecting more information on a job requires the allocation of the same resources used to process the job. This gives rise to operational tradeoffs of *exploration versus exploitation*, specifically, how to dynamically allocate resources between *diagnostic work* called *testing* that aims to collect more information on the arriving jobs, and *processing work* called *working* that simply serves the jobs (customers) in the systems. In this paper, we introduce new *scheduling models* that capture these tradeoffs, and provide some structural results and insights on the optimal policies as well as algorithmic results on how to obtain optimal and provably near-optimal policies. Surprisingly, in many interesting cases the optimal policy can be described through myopic (local) rules.

One relevant example for the type of tradeoff studied in this paper arises in *aircraft maintenance*. Engine repair requires disassembling and reassembling engines, which are costly in terms of time. Alternatively, engines can be diagnosed using special testing equipment, which can unveil the nature of breakdowns, and the required corrective measures and processing times. The shared resource in this case between working and testing are the maintenance personnel. Another example arises in emergency medical departments. In this setting, patients undergo the process of *triage* that aims at collecting information about their urgency (sensitivity to waiting), as well as the required activities and processing times. This information allows prioritizing patients to ensure efficient allocation of limited medical resources. While these examples stem from considerably different practices, they both give rise to similar tradeoffs, specifically, how resources should be allocated between diagnostics and actual processing of jobs.

The paper is focused on one of the core problems in scheduling theory, with a single server and the objective of minimizing the weighted sum of completion times of a given set of jobs. This objective reflects the goal of minimizing the weighted total (or average) wait time, which is realistic in the practical settings mentioned above and in many others, in which the tradeoff of testing versus working exists. The processing time and weight of the jobs is unknown but can be revealed if the job is *tested*, an activity that requires a specified server time. Thus, at any stage a decision has to be made whether to test another job or process a job (either one that was already tested, or one that still has uncertain processing time and weight.) Once a job is processed it has to be completed (i.e., preemption is not allowed.) We note that without the option to test, the problem is known to be solved optimally by processing jobs in an increasing order of their expected weighted processing times; this is known as the *Weighted Shortest Processing Time rule* (WSPT).

*Contributions.* This paper makes several important contributions. First, it introduces a new class of scheduling models that capture exploration versus exploitation tradeoffs in service environments. While it is widely recognized that understanding and controlling variability could be critical for

sustaining uninterrupted operations, to the best of our knowledge, this is the first work that studies the extent to which resources should be utilized to collect information and reduce uncertainty. Second, while a natural formulation of the problem leads to a high-dimensional *Dynamic Program* (DP), the paper provides structural analysis that obtains a characterization of optimal policies, which is managerially intuitive. Specifically, we explicitly identify (and compute) two thresholds that induce a partition of the tested jobs into three groups. The first group should be processed immediately with no delay. The second group should be processed last after all other jobs are processed. Finally, unknown jobs can be potentially tested only before known jobs from the third group are processed. We also show that the optimal policy has a structure of an optimal *stopping time* problem; it tests jobs and processes immediately jobs in the first group until at some point it switches to processing all remaining jobs using the WPST rule, and never tests again. Third, based on the structural characterization of the optimal policies together with an innovative *marginal cost accounting scheme*, we propose a low-dimensional DP formulation that unlike the natural high-dimensional one can be solved efficiently. Unlike traditional cost accounting schemes, in which the contribution of a job to the overall objective function is accounted for at the moment when the processing of this job is completed, in marginal cost accounting its contributions to the completion times of other jobs are accounted once the relative scheduling order between jobs is determined. Moreover, the structural properties of the optimal policy lead to a low-dimensional DP formulation that can be solved near-optimally for any specified degree of accuracy using a *fully polynomial approximation scheme* (FPTAS). Fourth, under a certain condition (which includes the special case of equally weighted jobs), the optimal policy is shown to be a *myopic rule* that can be based merely on current state. Fifth, the analysis provides insights into the value of testing as a function of the various parameters of the problem, as well as analytically assesses the performance of simpler policies. This analysis could be used to better understand and assess when the testing functionality is indeed worthwhile. Finally, the analysis extends to broader settings, in which testing might reveal only partial information about the latent attributes.

*Literature review.* For more than half a century the research community has developed a rich and extensive literature in the area of scheduling. Nevertheless, despite the wide spectrum of problems that has been explored, the topic of testing per se seems to have not receive attention. Typical features of scheduling problems concern aspects such as individual job properties (e.g., processing times, due dates, release dates and preemption), dependencies between jobs (e.g., precedence constraints, families of jobs, setup times), server properties (e.g., multiple servers, control of speed, batch processing, and breakdowns), server-job settings (e.g., flow shop, job shop, and open shop problems), under a variety of objectives (e.g., makespan, flow-time, lateness, and tardiness). The literature review of this paper does not attempt to present a comprehensive survey of this enormous

body of knowledge, but rather, present the main research areas within the scheduling literature, and their relation to our work. For a comprehensive treatment of the subject, the reader is referred to Pinedo (2012) and Leung (2004).

A principal way of classifying scheduling work is according to the amount of information known to the scheduler. The main categories are deterministic, stochastic, and online. The three decrease in the availability of information. In deterministic problems, all the information is known in advance, which implies that decisions can be made in advance, and that the overall performance can be predicted. Stochastic scheduling assumes a probabilistic characterization of uncertain data. On the far extreme lies online scheduling, where no knowledge regarding the processing or the arrival time to the system is assumed, and information is revealed gradually. The models studied in the paper share properties of both stochastic and online scheduling problems. On one hand, probability distributions are used to model uncertainties, but on the other hand, testing is allowed as a means of *learning* about these uncertainties in an online fashion.

Early work on deterministic scheduling focused on solving relatively simple problems, either by simple rules (such as the SPT or EDD rules, e.g., Smith (1956), Jackson (1955)), or using algorithms, often based on DP formulations, or branch and bound (e.g., Moore (1968), Lawler (1973), McMahon and Florian (1975)). Research then advanced to more complex systems, e.g., with multiple servers. While some of these problems could be solved optimally and efficiently (e.g., McNaughton (1959)), often they proved to be NP-hard (e.g., Bruno et al. (1974)). This line of research of proving hardness results of scheduling problems drew significant attention following the development of complexity theory (e.g., Lenstra et al. (1977), Lawler et al. (1993)). Later and to these days, the focus has shifted to approximation algorithms, that is, algorithms that sacrifice optimality to achieve computational efficiency. These are commonly based on solutions to relaxed versions of the original problem, often using LP (e.g., Hall et al. (1997), Chekuri et al. (2001)). One outcome of our work is an FPTAS for the problem (see Lawler (1982), Hochbaum and Shmoys (1987) for examples of FPTAS of specific models, and Hochbaum (1997), Chen et al. (1998), Ausiello et al. (1999) for an overview of approximation algorithms).

Research on stochastic scheduling tried, with various degrees of success, to solve stochastic counterparts of deterministic scheduling problems. While these models are often more realistic than the deterministic models, they also tend to be more difficult to solve. However, some results fully extend to the stochastic case (e.g., the WSEPT rule, Rothkopf (1966)), while others require special assumptions regarding the distribution functions (e.g., exponential distribution, Weiss and Pinedo (1980)). On certain occasions, stochasticity actually simplifies problems and their analysis (e.g., Pinedo (1983)). Usually however, the analysis of stochastic models is much harder, and consequently, much less is known about such systems. As in the deterministic case, significant

progress has been made on approximation algorithms for stochastic systems (e.g., Mohring et al. (1999)).

In the literature on maintenance, a notable effort has been devoted to studying maintenance problems with inspections, in a class of models that are known as preparedness models (McCall (1965)). In these models, machines deteriorate over time in a process that is hidden unless inspection is employed, which reveals the true state of the machine. Most of the related literature on inspection models focused on single-component systems in an infinite horizon setting, where breakdowns are invisible without inspections. Moreover, the assumptions are that inspections are costly and that their time is negligible, and the objective is to minimize costs. That is, the focus is on costs rather than on allocation of limited capacity. In multi-component systems, the work has mostly been on harnessing economies of scale to reduce maintenance costs by simultaneously repairing multiple components. In addition, models were created for systems in which a correlation exists between the evolution of components, or where components are jointly maintained due to structural dependencies. To the best of our knowledge, none of these works studied how inspections should be used to dynamically inform *scheduling* decisions. For surveys of maintenance models the reader is referred to: McCall (1965), Pierskalla and Voelker (1976), Sherif and Smith (1981), Sherif (1982), Yamayee (1982), Valdez-Flores and Feldman (1989), Cho and Parlar (1991), Dekker et al. (1997), Guo et al. (2000), Wang (2002), Frangopol et al. (2004), Jardine et al. (2006), Nicolai and Dekker (2008), and, van Noortwijk (2009).

We also note several results that study the value of information in a single server queuing and scheduling settings. Bansal (2005) studied an  $M/M/1$  queue, in which job durations are known upon arrival. He quantifies the improvement of a policy that processes jobs in an increasing order of the remaining processing time, over the standard first comes first served policy. Wierman and Nuyens (2008) studied a class of policies that generalizes the shortest processing time rule. These are used in practice when jobs with different processing times need to be grouped and assigned the same priority rule (which is similar to not having the exact information about the processing times). They derive bounds for multiple performance measures, and investigate how the bounds are affected by the accuracy of the information.

The tradeoff of exploration vs. exploitation has been studied in the context of several operational problems in revenue management and supply chain management (for example, see Besbes and Zeevi (2009) and Besbes and Muharremoglu (2013)). However, the typical assumption in this stream of work is that the underlying distributions are unknown and learnt from data. In contrast, in our setting the distributions are assumed known but specific instances from these distributions can be observed through testing.

The novelty of our work is in incorporating learning decisions into job scheduling problems. Traditionally, scheduling problems focused on determining the optimal sequence of jobs processing in a deterministic environment, or subject to uncertainties that are represented by probability distributions. However, to the best of our knowledge, the issue of testing has not been studied in published literature (the recent work of Sun et al. (2014) studied a very special case of this model).

The rest of this work is organized as follows. In Section 2 we describe the model, the new cost accounting scheme, and the resulting DP formulation. Section 3 contains an analysis of the model and the characterization of the optimal policy, a characterization that we then use in Section 4 to develop algorithms that solve the problem near-optimally. In Section 5 we study a myopic policy and prove its optimality under a certain assumption. In Section 6 we discuss the value of testing, and in Section 7 we generalize the model to a broader settings and show that the results of the basic model still hold. We conclude in Section 8 and discuss future research directions. Note that some of the proofs were omitted due to space limitation, but these appear in an online appendix.

## 2. Mathematical Formulation

Consider  $N_0$  jobs that need to be processed by a single non-preemptive server. Each job  $i$ , is associated with a given processing time  $t_i$  as well as a weight  $w_i$  that represents the relative importance of the job. The duration  $t_i$  and weight  $w_i$  of job  $i$  are a-priori random variables  $(T, W)$  distributed according to a joint distribution  $\mathcal{D}$  with support  $[1, D] \times [1, V]$ , and are independent and identically distributed across jobs.

When the server becomes idle, the scheduler could do one of the following. It can process a job, in which case the the processing time  $T$  and the weight  $W$  of the job are realized. However, the job must be processed with no preemption. Alternatively, the server can be used to **test** a job, which requires a specified processing time  $t_a$ , and reveals the required processing time and weight of the specific job. After testing, the job could be put on hold and processed later. Thus, whenever the server becomes idle, three decisions are available: process one of the “known” jobs (i.e., process a job that was tested), process an “unknown” job (i.e., process a job that was not tested), or test an “unknown” job. Note that both known and unknown pertain to jobs that have not yet been processed.

The system’s state can be expressed as a vector  $(N, [t_1, w_1, \dots, t_n, w_n])$ , where  $N$  and  $n$  denote the number of unknown and known jobs, respectively, and  $t_1, w_1, \dots, t_n, w_n$  denote the realization of the processing times and weights of each of the  $n$  known jobs. When there are no known jobs, the system state is simply  $(N, [])$ . Without loss of generality, we always assume that the ratio  $t_i/w_i$  is non-decreasing in  $i$ . We denote by  $\rho = \mathbb{E}[T]/\mathbb{E}[W]$  and  $\rho_i = t_i/w_i$  the processing time to weight (importance) ratio for an unknown job and a given tested job  $i$ , respectively. The action space

can be described by the set  $\{test, process_u, process_i\}$ . The controls refer to testing an unknown job, processing an unknown job, and processing the known job  $i$ . The goal is to find an adaptive scheduling policy that minimizes the expected weighted sum of completion times. This will be denoted as the S&T model (Scheduling with Testing).

Before presenting a DP formulation for the problem, we show in Section 2.1 below that a variant of the WSPT rule extends to our problem. This allows introducing a marginal cost accounting scheme in Section 2.2, which is then used to obtain a DP formulation for the problem (Section 2.3).

### 2.1. The WSPT Rule

The deterministic variant of the problem studied in this paper is known to be solved optimally by the policy that processes jobs in a non-decreasing order of their ratio (a.k.a., the WSPT rule or Smith’s rule, see Smith (1956)). A different (dynamic) view of this rule is that the optimal policy always selects for processing the job with the lowest ratio.

In this section, it is shown that a weaker version of this property holds for the S&T model (this will later be extended in Section 3). Specifically, we show that when processing, it is optimal to process a known job when its ratio is less than the ratio of an unknown job.

Note that while this property is usually proven using a simple interchange argument, in the S&T model test actions can take place between processing of any two jobs. In such a case, interchanging the two jobs to form a decreasing ratio order might no longer guarantee an improved scheduling policy. Moreover, by testing, we observe the true ratio of jobs, which might also affect the optimal scheduling order.

In Lemma 1 below, we show that given that two jobs that are processed consecutively, their ratio must be non-decreasing. We then prove in Lemma 2 a stronger property that given two jobs do not undergo testing, it is sub-optimal to process the job with the higher ratio before processing the job with the lower ratio (even when the jobs are not processed consecutively).

LEMMA 1. *It is sub-optimal to process a job immediately after processing a job with a higher ratio.*

*Proof.* See Appendix A.1.

LEMMA 2. *Processing a job (known or unknown) with a ratio higher than the ratio of a known job is sub-optimal.*

*Proof.* See Appendix A.2.

Note that Lemma 2 significantly reduces the actions space. At any state, we need to choose only between testing or processing an unknown job, and processing the known job with the smallest ratio.



## 2.2. Marginal Cost Accounting

Marginal cost accounting is related to the concept of Linear Ordering (see Queyranne et al. (1994) and the references therein). For the problem without testing, policies are described by linear ordering using the processing order of any pair of jobs. The objective value can then be written as  $\sum_{i=1}^n t_i w_i + \sum_{i \neq j} (1_{i < j} t_i w_j)$ , where  $1_{i < j}$  is the indicator function for the event that job  $i$  is processed before job  $j$ . We see that each job  $i$  contributes its processing time to itself ( $w_i t_i$ ), and to every job  $j$  that is processed afterward ( $t_i w_j$ ).

When jobs can be tested, one has to consider in addition the further delays caused by testing. For job  $i$ , the delays due to testing are  $t_a$  times the number of tested jobs prior to job  $i$ . With hindsight, we can write the objective value as follows:

$$J_{mrg}(N_0, []) = \sum_{i=1}^{N_0} t_i w_i + \sum_{i \neq j} (1_{i < j} t_i w_j) + \sum_{i=1}^{N_0} w_i t_a (\# \text{ of tested jobs prior to job } i).$$

Lemma 2 implies that as jobs are tested and their respective values  $t_i$ ,  $w_i$  become known, the optimal sequence of processing is partially determined. Therefore, some of the future costs can be computed at the time of testing. More generally, in marginal cost accounting, we charge all of the future costs that become known due to present action.

Specifically, when at state  $(N, [t_1, w_1, \dots, t_n, w_n])$  an unknown job  $l \notin \{1..n\}$  is tested and the values  $(t_l, w_l)$  are realized, then the delays caused by testing,  $t_a (\sum_{i=1}^n w_i + (N-1) \mathbb{E}[W] + w_l)$ , and the ordering costs with respect to other known jobs,  $t_l w_l + \sum_{i=1}^n (1_{\rho_l < \rho_i} (t_l w_i) + 1_{\rho_l > \rho_i} (t_i w_l))$  can be charged. Furthermore, if an unknown job is processed, the ordering costs with respect to all other jobs,  $\mathbb{E}[TW + \sum_{i=1}^n T w_i + (N-1) T \mathbb{E}[W]]$  can be charged. This include the “self imposing cost”  $TW$ , the costs associated with known jobs  $\sum_{i=1}^n T w_i$ , and the costs associated with the other  $N-1$  unknown jobs, which on expectation are  $(N-1) T \mathbb{E}[W]$  (we use the independence between jobs).

Finally, when known job 1 is processed, the additional costs are the ordering costs of job 1 with respect to the unknown jobs:  $N t_1 \mathbb{E}[W]$ . Note that other ordering costs have been already accounted for by the previous actions.

## 2.3. DP Formulation

In this section we describe a DP formulation of the problem. The state or the DP is represented by the vector  $(N, [t_1, w_1, \dots, t_n, w_n])$  defined previously. From Lemma 2 we can restrict the control space to  $\{test, process_u, process_1\}$ . The transitions are straightforward, whereas processing an unknown job decreases  $N$  by 1; processing job 1 removes job 1 from the state; testing an unknown job, decreases  $N$  by 1, and adds a known job to the state with the realizations of the processing time

and weight  $(T, W)$ . This occurs with probability defined by the distribution  $\mathcal{D}$ . Using the marginal cost account scheme (Section 2.2), we define the Bellman's equation as follows:

$$J_{mrg}(N, [t_1, w_1, \dots, t_n, w_n]) = \min \begin{cases} \mathbb{E}[TW] + (\sum w_i + N\mathbb{E}[W])t_a + & \text{test} \\ + \mathbb{E}[\sum_{i=1}^n \min\{Wt_i, w_iT\}] \\ + \mathbb{E}[J_{mrg}(N-1, [t_1, w_1, \dots, t_n, w_n] \cup \{T, W\})] \\ \mathbb{E}[TW] + (\sum w_i + (N-1)\mathbb{E}[W])\mathbb{E}[T] + & \text{process}_u \\ + J_{mrg}(N-1, [t_1, w_1, \dots, t_n, w_n]) \\ N\mathbb{E}[W]t_1 + J_{mrg}(N, [t_2, w_2, \dots, t_n, w_n]) & \text{process}_1 \end{cases} \quad (1)$$

$$J_{mrg}(0, [t_1, w_1, \dots, t_n, w_n]) = 0.$$

Since  $T$  and  $W$  are random, the transition to the next system state is random, and as a result, the cost-to-go is captured through expectation over all possible states. Note that this DP formulation has a high-dimensional state space that is likely to explode, making it computationally intractable to solve.

### 3. Properties of the Optimal Policy

In this section, the DP formulation of described in Section 2.3 is leveraged to characterize structural properties of optimal policies. These are then used to devise a low-dimensional DP formulation.

We start by introducing a new quantity  $\rho_a$ , that together with  $\rho = \mathbb{E}[T]/\mathbb{E}[W]$ , will be key to characterizing the optimal policy.

DEFINITION 1. The testing ratio  $\rho_a$  is defined as the unique solution to the equation:

$$\rho_a = x : t_a - \mathbb{E}[(xW - T)^+] = 0. \quad (2)$$

Lemma 3 below shows that  $\rho_a$  is well defined.

LEMMA 3. (1) The function  $f(x) = t_a - \mathbb{E}[(xW - T)^+]$  is non-increasing in  $x$ . Moreover,  $f(x)$  is strictly decreasing in  $x \geq \inf \text{Support}(T/W)$ , where its value is  $t_a > 0$ .

(2) The solution to  $t_a - \mathbb{E}[(xW - T)^+] = 0$  is unique.

(3) If  $x < \rho_a$  then  $t_a - \mathbb{E}[(xW - T)^+] > 0$ ; if  $x > \rho_a$  then  $t_a - \mathbb{E}[(xW - T)^+] < 0$ .

(4)  $\rho_a < \rho \Leftrightarrow t_a < \mathbb{E}[(\rho W - T)^+]$ .

*Proof.* Straightforward. Q.E.D.

The quantity  $\rho_a$  has the intuitive meaning of the minimal job ratio, for which testing earlier is favored to testing later. As we will see,  $\rho_a$  is a trigger point for testing unknown jobs; specifically, we will show that it is never optimal to test an unknown job after a known job  $i$  with  $\rho_a < \rho_i$  or

before a known job with  $\rho_a > \rho_i$ . Similarly,  $\rho$  will serve as a trigger point for processing unknown jobs, and we will show that it is never optimal to process unknown jobs after a known job  $i$  with  $\rho < \rho_i$ , or before a known job  $i$  with  $\rho > \rho_i$ .

Using  $\rho$  and  $\rho_a$  we can divide known jobs into three groups: (i) **low-ratio** jobs ( $\rho_i < \min(\rho, \rho_a)$ ); (ii) **medium-ratio** jobs ( $\min(\rho, \rho_a) < \rho_i < \max(\rho, \rho_a)$ ); and (iii) **high-ratio** jobs ( $\rho_i > \max(\rho, \rho_a)$ ). For ease of exposition we assume that  $\rho_a \neq \rho$  (no loss of generality), and that for each  $i$ , we have  $\rho_i \neq \rho_a$  and  $\rho_i \neq \rho$ . (This is with a slight loss of generality that can be easily resolved but hinders readability.) For state  $(N, [t_1, w_1, \dots, t_n, w_n])$ , denote the set of low/medium/high ratio jobs by  $\mathcal{S}_{Low}/\mathcal{S}_{Med}/\mathcal{S}_{High}$ , respectively. Observe that these sets are state-dependent.

Figure 1 illustrates this classification of jobs assuming that  $\rho_a < \rho$ . Jobs are ordered on the axis shown according to their ratio. Unknown jobs are denoted by a circle, and known jobs are denoted by “x”. In this example, there is one unknown job, one low-ratio job, three medium-ratio jobs, and two high-ratio jobs.

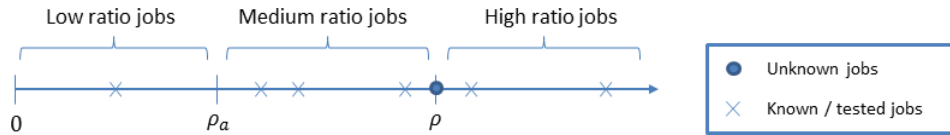


Figure 1 Classification of jobs according to their ratio (assuming  $\rho_a < \rho$ ).

The next lemma shows that the low-ratio jobs have the highest priority and are the first to be processed with no further delays immediately after they were tested.

LEMMA 4. *For any state  $(N, [t_1, w_1, \dots, t_n, w_n])$ , in which job 1 has a low-ratio, processing job 1 is the only optimal control.*

*Proof.* See Appendix A.3.

Lemma 4 implies that low-ratio jobs should be processed immediately upon testing. This implies that we can always assume that  $\rho_a < \rho_1$  and  $\rho < \rho_1$ . That is, under an optimal policy there is never a state with a low-ratio job.

We now consider two cases separately: (1)  $\rho_a < \rho$ , and (2)  $\rho_a > \rho$ . In the first case (Section 3.1), we show that any testing must precede any processing of medium-ratio, high-ratio, and unknown jobs. Consequently, all testing should be done immediately after processing low-ratio jobs, and once

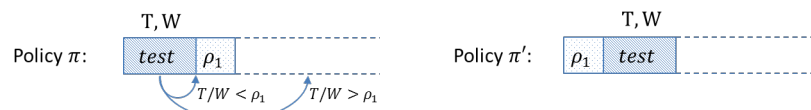


Figure 2 Example: testing before processing low-ratio jobs

we stop testing, all remaining jobs are processed in non-decreasing order of their ratio, essentially following the WSPT rule. For the second case (Section 3.2), we show that unknown jobs should never be tested. Therefore, the problem is reduced to the traditional problem (without testing), of minimizing the weighted sum of completion times.

### 3.1. Short Test Time ( $\rho_a < \rho$ )

Using the assumption that  $\rho_a < \rho$ , we first prove a local optimality condition that pertains to testing immediately after processing. Despite being local, this result together with the previous lemmas and properties impose a significant amount of structure on the optimal policy.

*LEMMA 5. If  $\rho_a < \rho$ , then testing immediately after processing jobs with ratio higher than  $\rho_a$  is sub-optimal.*

*Proof.* See Appendix A.4.

Lemmas 4 and 5 can be thought of as an interchange property of testing. Similarly to the way the interchange argument reorders jobs in the problem without testing, the two lemmas reorder testing so that it is performed after processing low-ratio jobs, and before processing any other job.

We conclude that when  $\rho_a < \rho$ , the optimal policy always processes all low-ratio jobs after testing, and generally operates in two phases. In the first phase, unknown jobs are tested, and processed immediately only if they have a low-ratio. In the second phase, all the jobs in the system are processed in a non-decreasing order of their ratio. This means that the problem can be seen as a stopping problem, where the decision to continue corresponds to testing an unknown job (and processing if the respective job has a low-ratio), and stopping corresponds to processing all remaining jobs (see Figure 3). These results are summarized in the following theorem.

*THEOREM 1. For  $\rho_a < \rho$ , the dynamics of the optimal policy are:*

- (1) Process all jobs with ratio below  $\rho_a$  in a non-decreasing order of their ratio;*
- (2) Either process all remaining jobs in a non-decreasing order of ratio, or, test a job and go back to (1).*

*(when we process all the jobs in non-decreasing order of their ratio (in case 2), we use  $\rho$  as the ratio for all unknown jobs and thus can process them all consecutively).*

*Proof.* Immediate from Lemmas 4 and 5. Q.E.D.

Interestingly enough, the form for the optimal solution bears a close resemblance to current practices of emergency departments. The highest priority is given to urgent patients (high weight), and to cases that can be quickly resolved (low processing times). Other cases are triaged (tested) and put on hold. This may suggest that the triage model should be considered in other industries (possibly adjusted to the specific area of application).

Note that some of the questions are yet unanswered. Mainly, should we test or process all jobs after all low-ratio jobs have been processed (that is, when to stop).

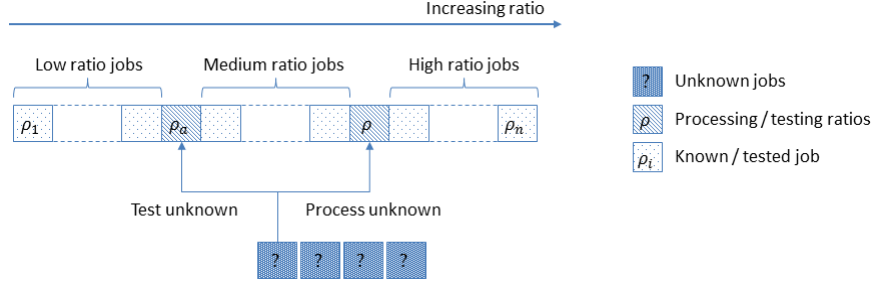


Figure 3 The structure of the optimal policy when  $\rho_a < \rho$ .

### 3.2. Long Test Time ( $\rho < \rho_a$ )

When  $\rho < \rho_a$ , we show that testing is always sub-optimal, which implies that the problem reduces to the traditional problem without testing.

**THEOREM 2.** *If  $\rho < \rho_a$ , then for every state  $(N, [t_1, w_1, \dots, t_n, w_n])$  the optimal policy processes all jobs in a non-decreasing order of their ratio (i.e., testing is never optimal).*

*Proof.* See Appendix A.5.

While a basic assumption of the model is that in the initial state of the system there are  $N_0$  jobs, Theorem 2 (and all other lemmas and theorems) holds even when the starting state contains known jobs.

Note that when  $\rho < \rho_a$ , the optimal policy of processing all jobs in a non-decreasing order of their ratios is a special case of the optimal policy when  $\rho_a < \rho$ , in which testing should not be performed.

## 4. Solutions and Algorithms

In this section, we develop an efficient algorithmic solution to the problem. In Section 4.1 we use the properties of the optimal policies proven in Section 3 to obtain a low-dimensional DP formulation. In Section 4.2 we analyze the new formulation, and in Section 4.3, we use it to develop an approximation scheme.

### 4.1. Low Dimensional DP Formulation

We start by defining several statistics for an arbitrary state  $(N, [t_1, w_1, \dots, t_n, w_n])$ :

- $\omega_M \triangleq \sum_{i \in S_{Med}} w_i$  (the total weight of medium-ratio jobs),
- $\omega_H \triangleq \sum_{i \in S_{High}} w_i$  (the total weight of high-ratio jobs),
- $\tau_M \triangleq \sum_{i \in S_{Med}} t_i$  (the total processing time of medium-ratio jobs)
- $\omega\tau \triangleq \sum_{i \in S_{Med} \cup S_{High}} \mathbb{E}[\min(Tw_i, t_i W)]$  (the expected ordering costs of a tested job and the known jobs).

Building on Theorem 1 and the stopping time interpretation for the S&T problem, we formulate an improved DP.

DEFINITION 2. The Low Dimensional DP is defined as following

$$J_{LD} \left( \begin{matrix} N, \\ \omega_M, \omega_H, \\ \tau_M, \omega\tau \end{matrix} \right) = \min \left\{ \begin{array}{l} \mathbb{E}[TW] + (\omega_M + \omega_H + N\mathbb{E}[W])t_a + \omega\tau + \\ \left( \begin{array}{l} 1_{\frac{d}{v} < \rho_a} d(N-1)\mathbb{E}[W] + \\ 1_{\frac{d}{v} < \rho_a} J_{LD}(N-1, \omega_M, \omega_H, \tau_M, \omega\tau) + \\ 1_{\rho_a < \frac{d}{v} < \rho} J_{LD} \left( \begin{matrix} N-1, \omega_M + v, \omega_H, \\ \tau_M + d, \\ \omega\tau + \mathbb{E}[\min(Tv, dW)] \end{matrix} \right) + \\ 1_{\rho < \frac{d}{v}} J_{LD} \left( \begin{matrix} N-1, \omega_M, \omega_H + v, \tau_M, \\ \omega\tau + \mathbb{E}[\min(Tv, dW)] \end{matrix} \right) \end{array} \right) \text{test-one} \\ NE[TW] + \mathbb{E}[W]N\tau_M + N\mathbb{E}[T]\omega_H + \binom{N}{2}\mathbb{E}[T]\mathbb{E}[W] \text{process-all} \end{array} \right. \quad (3)$$

$$J_{LD} \left( \begin{matrix} 0, \\ \omega_M, \omega_H, \\ \tau_M, \omega\tau \end{matrix} \right) = 0.$$

In these expressions,  $d$  and  $v$  are realizations of the processing time and weight of an unknown job.

There are only two controls in the LD DP: “test-one” and “process-all”. Testing one job refers to testing a job and processing the respective job if it has low-ratio. Processing all refers to processing all jobs in a non-decreasing order of their ratio.

We next show that there is an equivalence between the DP formulation of Section 2.3, and the LD DP.

THEOREM 3. *For every state  $(N, [t_1, w_1, \dots, t_n, w_n])$ , the following holds:*

$$J_{mrg}(N, [t_1, w_1, \dots, t_n, w_n]) = J_{LD}(N, \omega_M, \omega_H, \tau_M, \omega\tau)$$

*Proof.* Using marginal cost accounting (Section 2.2) and the characterization of the optimal policy (Section 3), we write the costs under the controls “test-one” and “process-all”. When we process all jobs, the entire processing order is determined. Figure 4 illustrates the costs for the control process-all. There are three types of costs arising from the job ordering: (1) pairs of unknown jobs  $\binom{N}{2}\mathbb{E}[T]\mathbb{E}[W] + N\mathbb{E}[TW]$  (Figure 4a); (2) pairs consisting of a medium-ratio jobs and unknown jobs  $\mathbb{E}[W]N \left( \sum_{i \in \mathcal{S}_{Med}} t_i \right)$  (Figure 4b); and (3) pairs of high-ratio jobs and unknown jobs  $N\mathbb{E}[T] \left( \sum_{i \in \mathcal{S}_{High}} w_i \right)$  (Figure 4c). The complete cost when we process all jobs is therefore:

$$N\mathbb{E}[TW] + \mathbb{E}[W]N \left( \sum_{i \in \mathcal{S}_{Med}} t_i \right) + N\mathbb{E}[T] \left( \sum_{i \in \mathcal{S}_{High}} w_i \right) + \binom{N}{2}\mathbb{E}[T]\mathbb{E}[W].$$

Similarly, Figure 5 illustrates the source of the costs for the control test one. In Figure 5a we see the system state before performing the test-one action. The immediate costs from the control

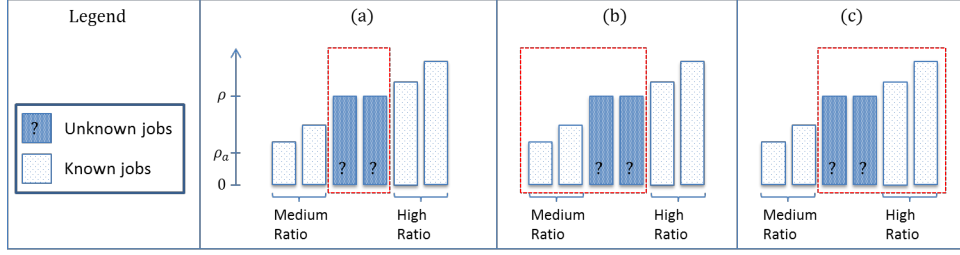


Figure 4 LD DP: cost incurred under the control process all.

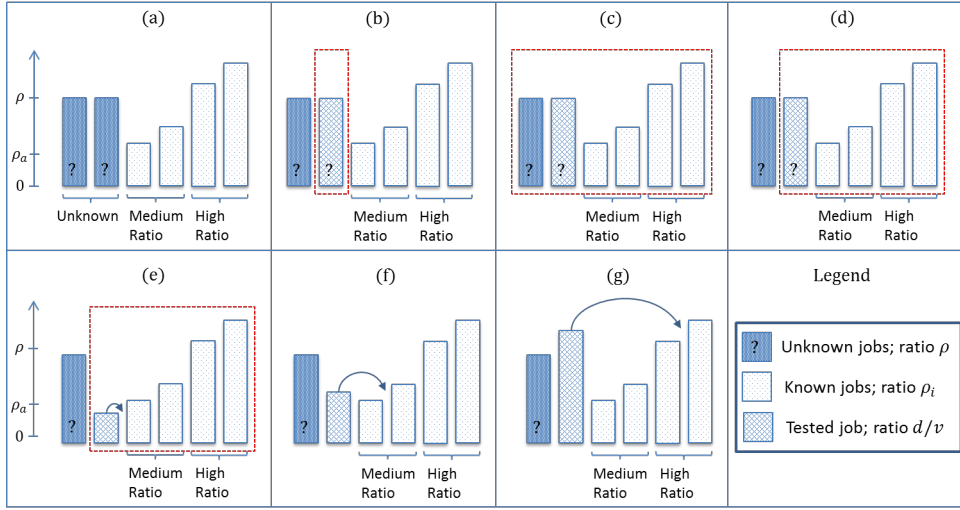


Figure 5 LD DP: cost incurred under the control test one.

include the self imposing costs of the tested job  $\mathbb{E}[TW]$  (Figure 5b), the costs from the testing delay  $\left(\sum_{i \in S_{Med} \cup S_{High}} w_i + N\mathbb{E}[W]\right)t_a$  (Figure 5c), and the costs induced by the pairs of known jobs and the tested job  $\sum_{i \in S_{Med} \cup S_{High}} \mathbb{E}[\min(Tw_i, t_i W)]$  (Figure 5d). Conditioning on the realization of the tested job, we add the costs-to-go. If the tested job has low-ratio (Figure 5e,  $d/v < \rho_a$ ), it is processed immediately and a cost of  $d(N-1)\mathbb{E}[W] + J_{mrg}(N-1, [t_1, w_1, \dots, t_n, w_n])$  is incurred. This is the processing time of the tested job multiplied by the number of unknown jobs and their expected weight, and the future costs from the next state, which is similar to the current state only with one less unknown job. Otherwise, if the tested job has medium-ratio (Figure 5f) or high-ratio (Figure 5g), we add the costs of the next state ( $J_{mrg}(N-1, [t_1, w_1, \dots, t_n, w_n] \cup \{d, v\})$ ) which contains one less unknown job, and an additional known job with processing time and weight  $(d, v)$ , respectively.

We can now write Bellman’s Equation for the DP with the controls “test-one” and “process-all”.

$$J_{mrg} \left( N, [t_1, w_1, \dots, t_n, w_n] \right) = \min \left\{ \begin{array}{l} \mathbb{E}[TW] + \left( \sum_{i \in \mathcal{S}_{Med} \cup \mathcal{S}_{High}} w_i + N\mathbb{E}[W] \right) t_a + \\ + \sum_{i \in \mathcal{S}_{Med} \cup \mathcal{S}_{High}} \mathbb{E}[\min(Tw_i, t_i W)] + \\ + \sum_{d, v \in \mathcal{S}_{\mathcal{D}}} p_{d, v} \left( \begin{array}{l} 1_{\frac{d}{v} < \rho_a} d(N-1)\mathbb{E}[W] + \\ 1_{\frac{d}{v} < \rho_a} J_{mrg}(N-1, [t_1, w_1, \dots, t_n, w_n]) + \\ 1_{\rho_a < \frac{d}{v} < \rho} J_{mrg} \left( \begin{array}{l} N-1, \\ [t_1, w_1, \dots, t_n, w_n] \\ \cup \{d, v\} \end{array} \right) + \\ 1_{\rho < \frac{d}{v}} J_{mrg} \left( \begin{array}{l} N-1, \\ [t_1, w_1, \dots, t_n, w_n] \\ \cup \{d, v\} \end{array} \right) \end{array} \right) \quad \text{test one} \\ \\ N\mathbb{E}[TW] + \mathbb{E}[W]N \left( \sum_{i \in \mathcal{S}_{Med}} t_i \right) + \\ + N\mathbb{E}[T] \left( \sum_{i \in \mathcal{S}_{High}} w_i \right) + \binom{N}{2} \mathbb{E}[T] \mathbb{E}[W] \quad \text{process all} \end{array} \right. \quad (4)$$

$$J_{mrg} \left( 0, [t_1, w_1, \dots, t_n, w_n] \right) = 0.$$

Observe that the DP cost function can be represented as a function of the number of unknown jobs ( $N$ ), the total weight of medium-ratio jobs ( $\omega_M = \sum_{i \in \mathcal{S}_{Med}} w_i$ ), the total weight of high-ratio jobs ( $\omega_H = \sum_{i \in \mathcal{S}_{High}} w_i$ ), the sum of processing times of medium-ratio jobs ( $\tau_M = \sum_{i \in \mathcal{S}_{Med}} t_i$ ), a summation over all known jobs ( $\omega\tau = \sum_{i \in \mathcal{S}_{Med} \cup \mathcal{S}_{High}} \mathbb{E}[\min(Tw_i, t_i W)]$ ), and a few other constants (e.g.,  $\mathbb{E}[TW]$ ). By substituting for these quantities, we obtain the equivalence of the two DP formulations. That is,

$$J_{mrg}(N, [t_1, w_1, \dots, t_n, w_n]) = J_{LD}(N, \omega_M, \omega_H, \tau_M, \omega\tau).$$

Q.E.D.

Note that the dimension of the state space is now five (comparing with the initial DP formulation in Section 2.3, where the dimension was as high as  $N_0$ ).

#### 4.2. Optimal Threshold Policy

The LD formulation has several advantageous properties in addition to having a low dimension. It is monotonous in its parameters, but more importantly, its optimal solution admits a threshold structure. That is, for every value of  $N, \omega_M, \omega_H$  and  $\tau_M$  there exists a threshold value for  $\omega\tau$  that determines if the optimal action is to test or to process all jobs.

LEMMA 6. *The value function  $J_{LD}$  is non-decreasing in  $\tau_M$  and  $\omega\tau$ .*

*Proof.* See Appendix A.9.



LEMMA 7. For every value of  $N, \omega_M, \omega_H$  and  $\tau_M$ , there exists a threshold value  $\overline{\omega\tau}$  such that testing is the optimal control, if and only if,  $\omega\tau \leq \overline{\omega\tau}$ .

*Proof.* See Appendix A.10.

Lemma 7 implies that there is an efficient way of representing the optimal policy when the support of the distribution  $\mathcal{D}$  is discrete. The number of different values that  $N, \omega_M, \omega_H$ , and  $\tau_M$  can take is polynomial in  $N_0, D$ , and  $V$ , and for every value of  $(N, \omega_M, \omega_H, \tau_M)$ , exactly one value of  $\overline{\omega\tau}$  is sufficient to describe the optimal policy.

However, to find the actual values of these thresholds we need to solve the DP. This cannot be done using standard DP methods due to the exponential growth of the state space. In the next section we develop an approximation scheme for solving the low-dimensional.

### 4.3. FPTAS

We use Approximate Dynamic Programming (ADP) to construct an FPTAS. We start by describing the state space of the ADP formulation. The state space is similar to the state space in the low-dimensional formulation, except for having rounded values for the second through the fifth dimensions. The first dimension of the state space takes values from the set  $\{0, 1, \dots, N_0\}$ . The second and third dimensions of the LD state space, namely  $\omega_M$  and  $\omega_H$ , can take values from the set  $\{\sum_{i \in S} w_i\}$ , where  $w_i \in \{1..V\}$ , and  $S$  represents a collection of known jobs. The minimal and maximal values of  $\omega_M$  and  $\omega_H$ , are 0 and  $N_0V$ , respectively. The discretized set of values is defined as:

$$\mathcal{S}_1 = \{0, 1, (1 + \delta), (1 + \delta)^2, \dots, \lceil N_0V \rceil\}.$$

The operator  $\lceil \cdot \rceil$  rounds up to the next power of  $1 + \delta$ . Similarly, the maximal values that the dimension  $\tau_M$  and  $\omega\tau$  can hold are  $N_0D$ , and  $N_0DV$ , respectively. The discretized sets of values for the dimensions  $\tau_M$  and  $\omega\tau$  are defined as:

$$\mathcal{S}_2 = \{0, 1, (1 + \delta), (1 + \delta)^2, \dots, \lceil N_0D \rceil\}, \mathcal{S}_3 = \{0, 1, (1 + \delta), (1 + \delta)^2, \dots, \lceil N_0DV \rceil\}.$$

Bellman's equation for the ADP can be written as follows:

$$J_\delta \left( \begin{array}{c} N, \\ \omega_M, \omega_H, \\ \tau_M, \omega\tau \end{array} \right) = \min \left\{ \begin{array}{l} \mathbb{E}[TW] + (\omega_M + \omega_H + N\mathbb{E}[W])t_a + \omega\tau + \\ \quad + \sum_{d,v \in \mathcal{S}_D} p_{d,v} \left( \begin{array}{l} 1_{\frac{d}{v} < \rho_a} d(N-1)\mathbb{E}[W] + \\ 1_{\frac{d}{v} < \rho_a} J_\delta(N-1, \lceil \omega_M \rceil, \lceil \omega_H \rceil, \lceil \tau_M \rceil, \lceil \omega\tau \rceil) + \\ 1_{\rho_a < \frac{d}{v} < \rho} J_\delta \left( N-1, \lceil \omega_M + v \rceil, \lceil \omega_H \rceil, \lceil \tau_M + d \rceil, \right. \\ \quad \left. \lceil \omega\tau + \mathbb{E}[\min(Tv, dW)] \rceil \right) + \\ 1_{\rho < \frac{d}{v}} J_\delta \left( N-1, \lceil \omega_M \rceil, \lceil \omega_H + v \rceil, \lceil \tau_M \rceil, \right. \\ \quad \left. \lceil \omega\tau + \mathbb{E}[\min(Tv, dW)] \rceil \right) \end{array} \right) + \text{test one} \quad (5) \\ \mathbb{E}[W]N\tau_M + N\mathbb{E}[T]\omega_H + N\mathbb{E}[TW] + \binom{N}{2}\mathbb{E}[T]\mathbb{E}[W] \quad \text{process all} \end{array} \right.$$

$$J_\delta \left( \begin{array}{c} 0, \\ \omega_M, \omega_H, \\ \tau_M, \omega\tau \end{array} \right) = 0.$$

The next lemma shows that the ADP can be used as a close approximation to the low-dimensional DP.

LEMMA 8. *For any state  $(N, \omega_M, \omega_H, \tau_M, \omega\tau)$ , the ratio between the value functions of the ADP and the LD DP formulations is greater than or equal to one, and at most  $1 + \delta$ :*

$$1 \leq \frac{J_\delta(N, \omega_M, \omega_H, \tau_M, \omega\tau)}{J_{LD}(N, \omega_M, \omega_H, \tau_M, \omega\tau)} \leq (1 + \delta)^N.$$

*Proof.* See Appendix A.11. Q.E.D.

The approximation algorithm is summarized in Algorithm 1. At each state, we use the ADP to make decisions in the low-dimensional DP problem space. The decision of testing or processing is done by evaluating the two controls using the approximated value function  $J_\delta$ . When at state  $(N, \omega_M, \omega_H, \tau_M, \omega\tau)$ , the decision we make is sub-optimal by a factor of at most  $(1 + \delta)^N$  (Lemma 8). Our next decision is sub-optimal by a factor of at most  $(1 + \delta)^{N-1}$ . On the overall, the approximation algorithm is sub-optimal by a factor of at most  $(1 + \delta)^{N^2}$ . Therefore, setting the value of  $\delta$  to  $e^{\frac{\log 1 + \epsilon}{N_0^2}} - 1$  ensures a  $1 + \epsilon$  approximation.

---

**Algorithm 1** Approximation Algorithm.

---

- 1: Choose  $\epsilon$
  - 2: Set  $\delta$  to  $e^{\frac{\log 1 + \epsilon}{N_0^2}} - 1$
  - 3: Solve the ADP: calculate  $J_\delta(N, \omega_M, \omega_H, \tau_M, \omega\tau)$  for  $N \leq N_0$ ,  $\omega_M, \omega_H \in \mathcal{S}_1$ ,  $\tau_M \in \mathcal{S}_2$ , and  $\omega\tau \in \mathcal{S}_3$
  - 4: Set the current state  $S$  to  $(N_0, 0, 0, 0, 0)$
  - 5: **while**  $S.N > 0$  **do**
  - 6:     Activate the control that minimizes the value function  $J_\delta(S)$
  - 7:     Update the current state based on the selected control and the observed realization
  - 8: **end while**
- 

The size of each of the sets  $\mathcal{S}_1, \mathcal{S}_2$ , and  $\mathcal{S}_3$  is polynomial in  $N_0$ ,  $\log D$ ,  $\log V$ , and,  $1/\log(1 + \delta)$ . By plugging the value of  $\delta$ , we obtain the following:

$$\frac{1}{\log(1 + \delta)} = \frac{N_0^2}{\log(1 + \epsilon)} = \mathcal{O}\left(\frac{N_0^2}{\epsilon}\right),$$

which means that the total number of states in the ADP formulation is polynomial in the input size and  $1/\epsilon$ , and we can solve the ADP in that same order of time.

The algorithm returns a  $1 + \epsilon$  approximation for the optimal policy, and its running time is polynomial in the input size and  $1/\epsilon$ . Therefore, it is an FPTAS.

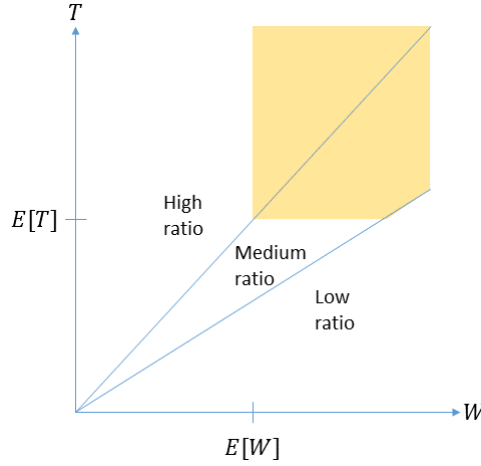


Figure 6 An illustration of Assumption 1.

## 5. Optimal Myopic Policies

The optimal and near-optimal algorithmic solutions presented in the previous section allow us to solve the problem in a polynomial number of steps. However, from a practical point of view,  $N^5$  may not be tractable for large instances. The heuristics on the other hand, can be efficiently implemented, but may not always have sufficiently good performance guarantees. In this section, we study a myopic policy that is both efficient and optimal under a relatively general assumption (which includes the case where all jobs have equal weights).

We start with a definition before stating the assumption and the main Theorem.

**DEFINITION 3.** The *Single test policy (STP)* is the policy in which a single unknown job is tested before processing all jobs in a non-decreasing order of their expected ratio (assuming there is at least one unknown job).

**LEMMA 9.** For any state  $(N, [t_1, w_1, \dots, t_n, w_n])$  with no low-ratio jobs and  $N > 0$ , the difference in the objective value between policies PA and STP is equal to:

$$\begin{aligned}
 J^{STP} - J^{PA} &= \left( N\mathbb{E}[W] + \sum_i w_i \right) t_a - (N-1)\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] \\
 &\quad - \sum_{i \in \text{Medium}} \mathbb{E}[(Wt_i - w_iT)^+] - \sum_{i \in \text{High}} \mathbb{E}[(w_iT - Wt_i)^+]
 \end{aligned} \tag{6}$$

*Proof.* See Appendix A.12.

**ASSUMPTION 1.** For all jobs  $i$  that have medium ratio:  $t_i \leq \mathbb{E}[T]$ , and for all jobs  $i$  that have high ratio:  $w_i \leq \mathbb{E}[W]$ .

Figure 6 illustrates distributions  $\mathcal{D}$  that satisfy Assumption 1. The support of such distributions reside outside of the shaded area. Note that Assumption 1 includes the case of equally weighted jobs  $w_i = \mathbb{E}[W]$ , where for medium-ratio jobs the condition  $t_i < \mathbb{E}[T]$  holds.

**THEOREM 4.** *Under Assumption 1, for any state  $(N, [t_1, w_1, \dots, t_n, w_n])$  with no low-ratio jobs and  $N > 0$ , the optimal control is to process all jobs, if and only if, the following condition holds:*

$$\begin{aligned} \left( N\mathbb{E}[W] + \sum_i w_i \right) t_a &\geq (N-1)\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] \\ &+ \sum_{i \in \text{Medium}} \mathbb{E}[(Wt_i - w_iT)^+] + \sum_{i \in \text{High}} \mathbb{E}[(w_iT - Wt_i)^+] \end{aligned} \quad (7)$$

*Proof.* See Appendix A.6.

The optimal policy under Assumption 1 is summarized in Algorithm 2. From a technical point of view, under Assumption 1 there is monotonicity in Equation (7) in the sense that with every test the difference between the lefthand side and righthand side increases. This allows us to identify the stopping condition. From an intuitive perspective, the optimal policy is sensitive to realizations within the shaded area. On expectation, it might not worth testing once, but if the realization happens to be an outcome with exceptionally high processing times and weight, we suddenly might want to test again since the cost of a sub-optimal schedule increased comparing to the testing cost. Based on this intuition, we construct an example and show that when Assumption 1 is not satisfied the myopic policy need not always be optimal (see Appendix A.7).

---

**Algorithm 2** The myopic policy algorithm.

---

- 1: Process all jobs with ratio below  $\rho_a$  in a non-decreasing order of their ratio.
- 2: **while** the following condition is satisfied

$$\begin{aligned} \left( N\mathbb{E}[W] + \sum_i w_i \right) t_a &< (N-1)\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] \\ &+ \sum_{i \in \text{Medium}} \mathbb{E}[(Wt_i - w_iT)^+] + \sum_{i \in \text{High}} \mathbb{E}[(w_iT - Wt_i)^+]; \end{aligned}$$

**do**

- 3: Test jobs and process them immediately if they have low-ratio.
  - 4: **end while**
  - 5: Process all jobs in a non-decreasing order of their ratio, where  $\rho$  is the ratio for all unknown jobs, and the processing order of any two jobs with equal ratio can be arbitrarily chosen.
- 

## 6. The Value of Testing

In this section we address the question of how much can we actually gain from testing. We start by analyzing a few simple heuristics, and then examine how the different problem parameters affect the value of testing.

## 6.1. Heuristics

We analyze the performance of three simple policies:

- “Process-all” ( $PA$ , which was introduced in Section 3.1),
- “Test-all first” ( $TAF$ ),
- “Test-all process-low-ratio” ( $TAPL$ ).

As their names suggest, in “test-all first”, we start by testing all jobs, and then process them in a non-decreasing order of job ratios. Under the policy “test-all process low-ratio”, we test all unknown jobs, but immediately process low-ratio jobs, and process other known jobs in a non-decreasing order of ratio only after all testing has been completed. Note that policies  $PA$  and  $TAPL$  correspond to the two extremes of the optimal policy: the first, when the stopping time is zero, and the latter, when the stopping time is  $N$ . We denote by  $OPT$  the optimal policy.

**6.1.1. Clairvoyant Solution (CL)** We use the clairvoyant solution ( $CL$ ) as a lower bound for the optimal policy. That is, we calculate the objective value when the processing times and weights are known to the scheduler (or alternatively if the testing time is zero). Using marginal cost accounting, the objective value is

$$\begin{aligned} J^{CL}(N, \square) &= \mathbb{E} \left[ \sum_{i=1}^N T_i W_i + \sum_{i=1}^N \sum_{j=1}^{i-1} \min(W_i T_j, W_j T_i) \right] \\ &= N \mathbb{E}[TW] + \binom{N}{2} \mathbb{E}[\min(W_i T_j, W_j T_i)]. \end{aligned}$$

The objective value consists of the self imposing costs ( $\sum_{i=1}^N T_i W_i$ ), and the ordering costs of the perfectly ordered jobs (as all the information is known in advance).

**6.1.2. The “Process All” (PA) Policy** Like the clairvoyant solution, we use marginal cost accounting to calculate the objective value under policy  $PA$ :

$$J^{PA}(N, \square) = N \mathbb{E}[TW] + \binom{N}{2} \mathbb{E}[T] \mathbb{E}[W].$$

The self imposing cost of every job is  $\mathbb{E}[TW]$  and every pair contributes a cost of  $\mathbb{E}[T] \mathbb{E}[W]$  to the objective (we use the independence between jobs). We obtain the following bound on the objective value of policy  $PA$ :

$$\begin{aligned} \frac{J^{PA}(N, \square)}{J^{OPT}(N, \square)} &\leq \frac{J^{PA}(N, \square)}{J^{CL}(N, \square)} \\ &= \frac{N \mathbb{E}[TW] + \binom{N}{2} \mathbb{E}[T] \mathbb{E}[W]}{N \mathbb{E}[TW] + \binom{N}{2} \mathbb{E}[\min(W_i T_j, W_j T_i)]} \\ &\leq \frac{\binom{N}{2} \mathbb{E}[T] \mathbb{E}[W]}{\binom{N}{2} \mathbb{E}[\min(W_i T_j, W_j T_i)]} \\ &= \frac{\mathbb{E}[T] \mathbb{E}[W]}{\mathbb{E}[\min(W_i T_j, W_j T_i)]}. \end{aligned}$$

The process all policy is of special interest to us, as it serves as basis for comparison when testing is not available. Put another way, comparing the optimal policy with the process all policy tells us the value of testing.

To get a better sense of the bound, we list below its values for a sample of distributions for the unweighted case ( $W = 1$ ). In favor of readability, the proofs are included in Appendix A.13.

LEMMA 10. *If  $T \sim Uni(a, b)$  and  $W = 1$  the policy  $PA$  is a  $\frac{3(b+a)}{2(b+2a)}$ -approximation for the optimal policy.*

LEMMA 11. *If  $T \sim Exp(\lambda)$  and  $W = 1$  the policy  $PA$  is a 2-approximation for the optimal policy.*

LEMMA 12. *If  $T \sim \begin{cases} a & p \\ b & 1-p \end{cases}$  and  $W = 1$  the policy  $PA$  is a  $\frac{pa+(1-p)b}{a+(1-p)^2(b-a)}$ -approximation for the optimal policy.*

In the last example, when  $a = 0, p = 1/2$ , policy  $PA$  is a 2-approximation, and for  $a = 0, b = M, p = 1 - 1/M$ ,  $PA$  is an  $M$ -approximation. This means that policy  $PA$  can do arbitrarily bad. When the testing time is relatively short, the clairvoyant solution is close to the optimal policy, and the bound becomes tight. In these examples, we see that testing can improve the objective by 33% for the uniform distribution, by 50% for the exponential distribution, and in some cases can do even better (e.g., the  $M$ -approximation).

**6.1.3. The “Test-All First” (TAF) Policy** When the testing time is short, it makes sense to test all jobs before doing any processing so that processing will be performed in the optimal order. It is easy to see that:

$$J^{TAF}(N, \square) = t_a N^2 \mathbb{E}[W] + J^{CL}(N, \square), \quad (8)$$

which results in an approximation ratio of

$$\begin{aligned} \frac{J^{TAF}(N, \square)}{J^{OPT}(N, \square)} &\leq \frac{J^{TAF}(N, \square)}{J^{CL}(N, \square)} \\ &= \frac{t_a N^2 \mathbb{E}[W] + J^{CL}(N, \square)}{J^{CL}(N, \square)} \\ &\leq 1 + \frac{2t_a \mathbb{E}[W]}{\mathbb{E}[\min(W_i T_j, W_j T_i)]}. \end{aligned}$$

Indeed, as  $t_a$  approaches zero, the  $TAF$  policy becomes optimal.

**6.1.4. The “Test-All Process Low-Ratio” (*TAPL*) Policy** One could try to improve policy *TAF* by processing low-ratio jobs immediately upon detection (Lemma 2).

The *TAPL* policy has two types of additional costs compared with the clairvoyant solution: due to testing delays, and due to sub-optimal processing order. In terms of testing delays costs, the completion time of all medium- and high-ratio jobs include the testing time of all  $N$  jobs, which results in an additional cost of  $t_a \mathbb{E}[W|T/W > \rho_a] N$  per job. Low-ratio jobs carry average testing delay costs of  $t_a \mathbb{E}[W|T/W < \rho_a] (1 + N)/2$ .

In terms of ordering costs, medium- and high-ratio jobs are scheduled optimally with respect to all other jobs. Low-ratio jobs are optimal with respect to medium- and high-ratio jobs, but could be sub-optimal with respect to other low-ratio jobs. This yields an expected additional cost of  $\mathbb{E}[(W_2 T_1 - W_1 T_2)^+ | T_1/W_1, T_2/W_2 < \rho_a]$  to every pair of low-ratio jobs.

Denoting by  $N^l$  the number of low-ratio jobs, we obtain the following:

$$J^{TAPL}(N, \square) = J^{CL}(N, \square) + \mathbb{E} \left[ \binom{N^l}{2} \right] \mathbb{E} \left[ (W_2 T_1 - W_1 T_2)^+ \mid \frac{T_1}{W_1}, \frac{T_2}{W_2} < \rho_a \right] \\ + t_a \mathbb{E}[N - N^l] N \mathbb{E} \left[ W \mid \frac{T}{W} > \rho_a \right] + t_a \mathbb{E}[N^l] \frac{1 + N}{2} \mathbb{E} \left[ W \mid \frac{T}{W} < \rho_a \right]. \quad (9)$$

Comparing the costs under policies *TAF* and *TAPL*, we observe that under policy *TAF*, the total testing time penalty ( $t_a N^2 \mathbb{E}[W]$ ) is higher, but on the other hand, the processing order is optimal. In other words, policy *TAPL* trades the optimality of the processing order, in exchange for lower testing times. Using Equations (8) and (9), we can conclude in what cases the tradeoff is worthwhile.

## 6.2. Variability

In order to obtain insights on how the variability of the distribution  $\mathcal{D}$  affects the value of testing, we consider a more simple case without weights. We compare the model with the distribution of  $T$  with a more variable random variable  $T'$  from a distribution that admits convex ordering with respect to  $T$ , that is  $T \leq_{cx} T'$ .

In this case the bound on the process all policy becomes  $\mathbb{E}[T]/\mathbb{E}[\min(T_j, T_i)]$ . Since the function  $\min$  is concave, it is easy to see that  $\mathbb{E}[\min(T_j, T_i)] \geq \mathbb{E}[\min(T'_j, T'_i)]$ , and therefore

$$\frac{\mathbb{E}[T]}{\mathbb{E}[\min(T_j, T_i)]} \leq \frac{\mathbb{E}[T']}{\mathbb{E}[\min(T'_j, T'_i)]}.$$

As variability increases, the bound becomes looser. This seems to indicate that the value of testing increases with variability. It matches our intuition since that when variability is higher, the processing order of jobs becomes more significant (e.g., the processing order of jobs with duration 1, 10 is more critical than the processing order of jobs with duration 5, 6).

### 6.3. Sequential Tests

We turn to investigate how the information obtained from testing, affects the value of future tests. Some intuition can be gained from analyzing the low-dimensional DP formulation. Specifically, we perform a sensitivity analysis to learn how changes in the values of  $t_i$  and  $w_i$  affect the optimal control. We look at the derivatives of the value function under each of the controls *test one* and *process all*.

LEMMA 13. *The derivatives of the value function under the controls test and process all with respect to  $w_i$  and  $t_i$  satisfy the following conditions:*

$$\begin{aligned}
 (1) \quad \frac{d}{dt_i} J_{LD}^{process} &= \begin{cases} N\mathbb{E}[W] & i \in \mathcal{S}_{Med}; \\ 0 & i \in \mathcal{S}_{High}; \end{cases} \\
 (2) \quad \frac{d}{dw_i} J_{LD}^{process} &= \begin{cases} 0 & i \in \mathcal{S}_{Med}; \\ N\mathbb{E}[T] & i \in \mathcal{S}_{High}; \end{cases} \\
 (3) \quad \frac{d}{dt_i} J_{LD}^{test} &\leq \frac{d}{dt_i} \mathbb{E}[\min(w_i T, W t_i)] + (N-1)\mathbb{E}[W]; \\
 (4) \quad \frac{d}{dw_i} J_{LD}^{test} &\leq N t_a + \frac{d}{dw_i} \mathbb{E}[\min(w_i T, W t_i)] + (N-1)\mathbb{E}[T].
 \end{aligned}$$

*Proof.* Straightforward from Equation (3). Q.E.D.

Considering a medium-ratio job  $i$ , Lemma 13 indicates that if the processing time of the respective job increases, testing will become more advantageous compared to processing (since  $\frac{d}{dt_i} J_{LD}^{process} > \frac{d}{dt_i} J_{LD}^{test}$ ). On the other hand, if job  $i$  has high-ratio and its processing time increases, then testing becomes less attractive. That is, testing becomes more attractive when the changes draw the ratio of a known job towards  $\rho$ . We observe a similar insight, when the weight of medium-ratio jobs decreases, which increases jobs ratios, and makes testing more advantageous. This suggests that the test action is optimal when the ratio of known jobs is closer to  $\rho$ , and processing all jobs is optimal when the known job ratios are far from  $\rho$ .

### 6.4. The Initial Number of Jobs

We proceed to study how the value of testing is affected by the initial number of unknown jobs. Specifically, we use a sufficient condition on the state space for which testing is optimal, to find a lower bound on the number of times that the control *test one* is optimal (assuming  $\rho_a < \rho$ ).

The following quantity is important for describing the lower bound on the optimal stopping time.

DEFINITION 4. We define the *stopping factor*  $\beta$  as

$$\beta := \frac{t_a^{max}}{t_a} = \frac{\mathbb{E}[(\rho W - T)^+]}{\mathbb{E}[(\rho_a W - T)^+]} = \frac{\mathbb{E}[(\mathbb{E}[T] W - T\mathbb{E}[W])^+]}{t_a \mathbb{E}[W]}.$$

To develop intuition regarding the value  $\beta$ , consider the case when an unknown job is tested immediately after processing another unknown job. The stopping factor  $\beta$  is the ratio between the savings from testing earlier (using an improved schedule), and the additional cost of testing



(from having another job waiting while we test). Intuitively, the higher the values of  $\beta$ , it is more beneficial to test. (Note that as the testing time decreases,  $\beta$  increases, which matches our intuition that for higher values of  $\beta$ , testing is more beneficial.) Also note that when the ratio is less than 1, it is sub-optimal to test (since  $\rho_a < \rho$ ). We can therefore assume that  $\beta$  is greater than 1.

LEMMA 14. *For every state  $(N, \omega_M, \omega_H, \tau_M, \omega\tau)$ , where  $\beta > (N\mathbb{E}[W] + \omega_H) / (N - 1)$ , the optimal control is to test.*

*Proof.* See Appendix A.14. Q.E.D.

Note that the lefthand side of the inequality in Lemma 14 above (i.e.,  $\beta$ ) is a constant for the problem, and that the righthand side dynamically evolves as a function of the state. Moreover,  $\beta > 1$ , and the righthand side goes to 1 as  $N$  becomes larger. Therefore, when the number of initial unknown jobs is large, testing is optimal for an increasing number of jobs. From a practical point of view, in congested systems with many unknown jobs, testing is most beneficial.

In the next theorem, we use Lemma 14 to obtain a lower bound on the minimal number of tests.

THEOREM 5. *For finite and symmetrical distributions of jobs weight, the optimal policy tests for at least  $N_{tests}$  periods where*

$$N_{tests} = \left\lfloor N_0 \frac{\beta - 1}{\beta + 1} - \frac{\beta}{\beta + 1} \right\rfloor = N_0 - 1 - \left\lceil \frac{2N_0 + \beta}{\beta + 1} \right\rceil.$$

*Proof.* See Appendix A.8. Q.E.D.

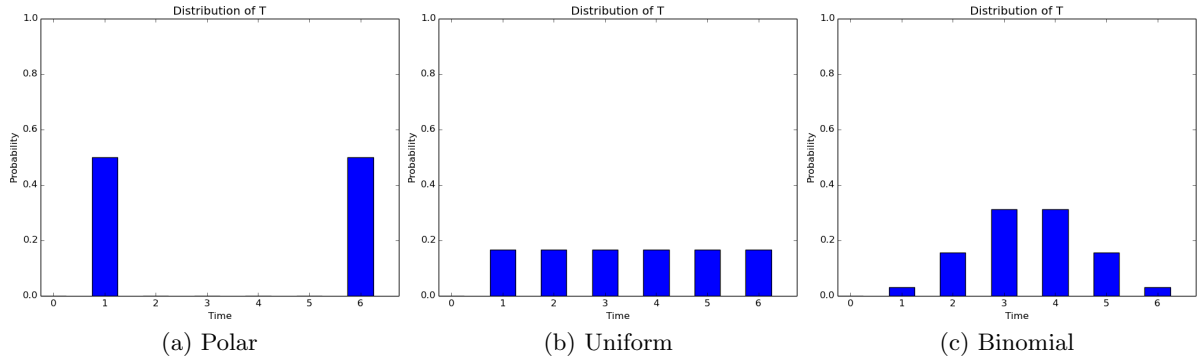
Lemma 14 suggests that the higher the number of unknown jobs is, the more likely we are to test, that is the value of information increases with  $N_0$ .

Another implication of Lemma 14 (that can be derived similarly to Theorem 5), is that when the starting state has  $N_0$  unknown jobs, to find the optimal policy we need to solve a DP with only  $N = \beta / (\beta - 1)$  jobs. While this quantity could be high (especially when the testing time is short), it does not depend on  $N_0$ , which indicates that the policy that tests all  $N_0$  jobs and processes them if they have low-ratio is asymptotically optimal (to see this, we briefly note that  $J_{LD}(N_0, 0, 0, 0, 0) = \Theta(N_0^2)$ , and that  $J_{LD}(\beta / (\beta - 1), \omega_M, \omega_H, \tau_M, \omega\tau) = \Theta((\beta / (\beta - 1))^2 + (\beta / (\beta - 1))N_0)$ , which means that testing and processing low-ratio jobs immediately can be sub-optimal up to a cost that decreases asymptotically with  $N_0$ ).

## 6.5. The Testing Time

Using the condition  $\rho < \rho_a$ , we derive a threshold value for the testing time above which testing is never optimal.

THEOREM 6. *If the testing time is greater than  $t_a^{max} := \mathbb{E}[(\rho W - T)^+]$ , testing is never optimal.*



**Figure 7** The probability mass function of the Polar, Uniform, and, Binomial distributions.

*Proof.* By definition, if  $t_a > t_a^{max}$ , then  $t_a > \mathbb{E}[(\rho W - T)^+]$ . Therefore,  $\rho_a > \rho$  (Lemma 3), and testing is not optimal (Theorem 2). Q.E.D.

Theorem 6 provides an upper bound on the testing time for which testing is still beneficial. In Section 6.4 we saw that it is in fact the smallest possible bound (when  $\rho_a < \rho$ , testing is always optimal for large enough values of  $N_0$ ).

### 6.6. Numerical Illustration

To illustrate the aforementioned observations, we plot in Figure 8 the improvement to the objective value from testing, for three probability distributions (see Figure 7), and three values of the initial number of jobs ( $N_0 = 6, 7, 8$ ).

For all curves, we see that when the testing time is high, processing all jobs is optimal (that is the value of testing is zero, Theorem 6). As the testing time decreases, the performance of policy  $PA$  deteriorates, and the optimal policy is a two-phase policy that transitions between  $TAPL$  to  $PA$ . Policy  $PA$  is worst when  $t_a \rightarrow 0$ , where the optimal policy coincides with policies  $TAF$  and  $TAPL$ . We also see in the figure, that the value of information increases with the number of initial jobs, and the variance of the distribution  $\mathcal{D}$  (with the variance of the polar distribution being the highest, and the binomial distribution being the lowest).

## 7. Extensions

In this section, we study a generalization of the problem and show that it can be solved using the analysis and methods presented in Sections 3 and 4. Specifically, we consider the case where testing does not reveal the exact processing times and weights, but rather the class of the respective job. For example, in the context of emergency departments, testing may reveal the type of treatment required by the patient but there may still be uncertainty associated with the actual service time, and severity.

We assume that there are  $C$  classes, and that for class  $i \in C$ , the processing time and weight of every job  $T_i, W_i$  are R.V.s from a known distribution  $\mathcal{D}_i$  with expected processing time  $\bar{T}_i$ ,

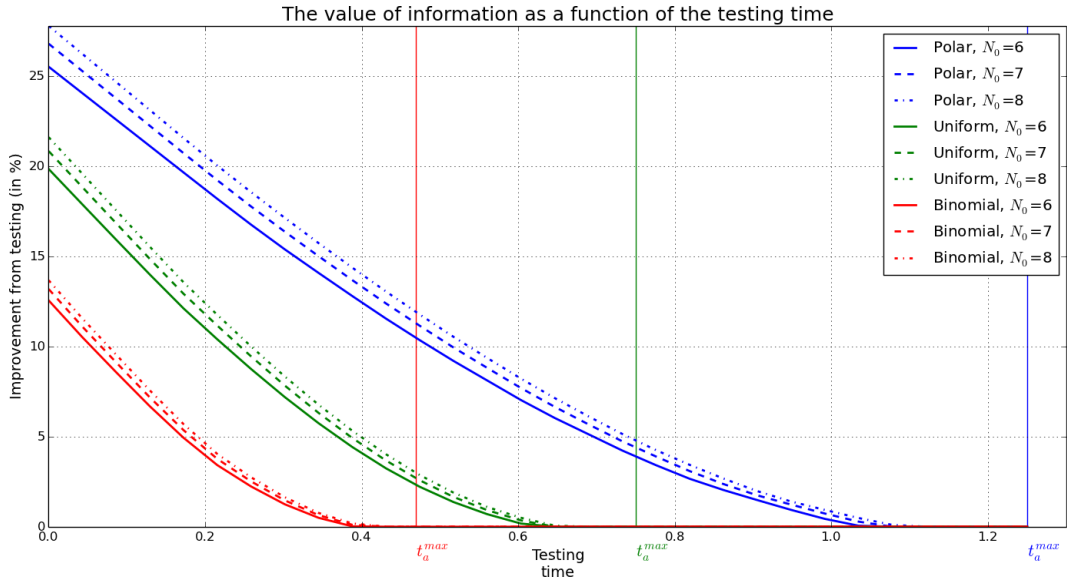


Figure 8 The value of testing for different distributions and initial number of jobs.

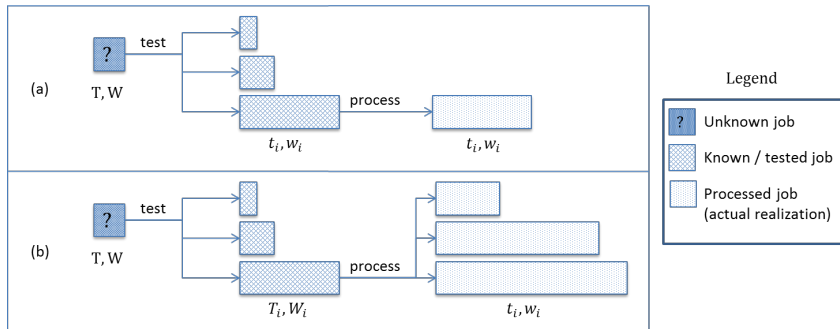


Figure 9 The information revealed by testing in the original problem (a), and in the generalized problem (b).

and expected weight  $\bar{W}_i$ . The probability that a job belongs to class  $i$  is denoted by  $p_i^c$  (which implies that the expected processing time and weight of an unknown job are  $\mathbb{E}[T] = \sum p_i^c \bar{T}_i$ , and  $\mathbb{E}[W] = \sum p_i^c \bar{W}_i$  respectively). Testing a job reveals its class and requires  $t_a$  units of time. Here we denote by unknown, jobs that were not yet tested, and by known jobs that were tested and which class is known (although actual processing time and weight may still be random).

Figure 9 illustrates the differences between the original and generalized problems. In the original problem (9a) the exact processing time is known once testing is performed. On the other hand, in the generalized problem (9b), only the class is realized by testing, and the exact processing time is known only after processing. Note that in the generalized problem, jobs can be tested at most once, that is, we can test a job to find its class and the respective distribution  $\mathcal{D}_i$  of the processing time and weight, but we cannot test it further for exact values. Also note, that if the distributions  $\mathcal{D}_i$  are degenerate, the extended problem reduces to a discrete instance of the original problem.

While the generalized problem is more complex, the analysis of Section 3 carries through. To prove this, we take advantage of two properties: (1) the linearity of the expectation operator and the objective function; and (2) the independence between jobs. In what follows, we discuss how to modify the proofs and algorithms to accommodate the general model.

The definition of jobs ratio perfectly extends to the generalized model, with the ratio of a known job  $i$  being  $\rho_i = \mathbb{E}[T_i]/\mathbb{E}[W_i]$ . We define the processing and testing ratios in the same way. Lemma 1 applies regardless to whether jobs are deterministic or stochastic, and it is easy to see that Lemma 2 still holds if we replace the quantities  $t_1, w_1$  with the random variables  $T_1, W_1$  and take expectation. Therefore, we can once again limit the control set to testing or processing the job with the smallest ratio.

The DP formulation for the extended problem can be developed similarly to the original formulation (Section 2.3). The state space can be described by  $N$ , the number of unknown jobs, and by  $\bar{T}_1, \bar{W}_1, \dots, \bar{T}_n, \bar{W}_n$  which are the expected values of the random variables corresponding to the tested jobs (similarly,  $\rho_i$  is non-decreasing in  $i$ ). The extended DP formulation can now be written as following:

$$J_{ext}(N, [\bar{T}_1, \bar{W}_1, \dots, \bar{T}_n, \bar{W}_n]) = \min \begin{cases} \mathbb{E}[TW] + (\sum \bar{W}_i + N\mathbb{E}[W])t_a + & \text{test} \\ + \mathbb{E}[\sum_{i=1}^n \min\{W\bar{T}_i, \bar{W}_i T\}] \\ + \mathbb{E}[J_{ext}(N-1, [\bar{T}_1, \bar{W}_1, \dots, \bar{T}_n, \bar{W}_n] \cup \{T, W\})] \\ \mathbb{E}[TW] + (\sum \bar{W}_i + (N-1)\mathbb{E}[W])\mathbb{E}[T] + & \text{process}_u \\ + J_{ext}(N-1, [\bar{T}_1, \bar{W}_1, \dots, \bar{T}_n, \bar{W}_n]) \\ N\mathbb{E}[W]\bar{T}_1 + J_{ext}(N, [\bar{T}_2, \bar{W}_2, \dots, \bar{T}_n, \bar{W}_n]) & \text{process}_1 \end{cases}$$

$$J_{ext}(0, [\bar{T}_1, \bar{W}_1, \dots, \bar{T}_n, \bar{W}_n]) = 0.$$

The value function for the control process unknown, and process job 1 can be derived using the fact the known jobs are independent from of each other (e.g.,  $\mathbb{E}[WT_1] = \mathbb{E}[W]\mathbb{E}[T_1]$ ). For the test control, observe that the job ordering cost (i.e.,  $\mathbb{E}[\min\{W\bar{T}_i, \bar{W}_i T\}]$ ) between the tested job and the already known job  $i$ , is achieved as following. If the the ratio of the tested job ( $T/W$ ) is smaller than the ratio of job  $i$  (i.e.,  $W\bar{T}_i > \bar{W}_i T$ ), a cost of  $\bar{W}_i T$  is added. Otherwise, if  $W\bar{T}_i < \bar{W}_i T$ , we add the cost  $W\bar{T}_i$ . This is equivalent to adding the cost  $\min\{W\bar{T}_i, \bar{W}_i T\}$ . The other components of the test control are derived similarly to the original problem.

Considering the DP formulation for the extended problem, observe that it is identical to a discrete instance of the original problem, where jobs are realized to their expected values. This means that we can find the value function of the extended problem using the solution of an instance of the original problem. Practically speaking, the extended problem reduces to the original problem.

This implies that all the properties from Section 3 still hold, and that we can use the algorithms developed in Section 4 to solve the extended problem efficiently.

We conclude this section by noting that while a basic assumption of the model was that the testing time is a constant, it is easy to show that results follow even when the testing time is a random variable  $T_a$ , which may be correlated with the processing time and weight of the tested job. In this case, the testing threshold  $\rho_a$  is similarly defined using the expectation of  $T_a$  (instead of using  $t_a$ ). A small modification to the DP formulation of Section 2.3 is needed, specifically, replacing the expression  $Nt_a\mathbb{E}[W]$  by the term  $(N-1)\mathbb{E}[T_a]\mathbb{E}[W] + \mathbb{E}[WT_a]$ . We also note that if the testing time is part of the processing time, that is, testing decreases the processing time by  $t_a$ , the structure of the optimal policy is preserved. In this case however, the testing threshold  $\rho_a$  is smaller than the threshold of the original problem.

## 8. Conclusions

In this paper we introduced a new class of models that captures a principal exploration-exploitation tradeoff that is common in many scheduling problems. We analyzed the problem and found an intuitive characterization of the optimal policy. For a large number of cases, the optimal policy was given explicitly in the form of a stopping rule. For all other cases, a novel cost accounting scheme was used to formulate a low-dimension DP, which lead to optimal and near optimal algorithms. We studied the performance of several intuitive policies, and how the problem parameters affect the value of testing. Finally, it was shown that the properties and algorithms extend to a more general model.

The concept of testing in scheduling problems seems relevant in many other setting. Some of the directions that seem interesting to pursue are more general scheduling models with arrival of jobs and multiple servers, models where the degree of testing can be controlled, and tests that reveal additional information about the jobs, such as their due dates or the arrival times.

## References

- Ausiello, G., P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi. 1999. *Complexity and Approximability Properties: Combinatorial Optimization Problems and Their Approximability Properties*. Springer.
- Bansal, Nikhil. 2005. On the average sojourn time under M/M/1/SRPT. *Operations Research Letters* **33**(2) 195–200.
- Besbes, Omar, Alp Muharremoglu. 2013. On implications of demand censoring in the newsvendor problem. *Management Science* **59**(6) 1407–1424.
- Besbes, Omar, Assaf Zeevi. 2009. Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research* **57**(6) 1407–1420.

- Bruno, J.1, Jr. Coffman, R.1 Sethi. 1974. Scheduling independent tasks to reduce mean finishing time. *Communications of the ACM* **17**(7) 382–387.
- Chekuri, C., R. Motwani, B. Natarajan, C. Stein. 2001. Approximation techniques for average completion time scheduling. *SIAM Journal on Computing* **31**(1) 146–166.
- Chen, Bo, Chris N. Potts, Gerhard J. Woeginger. 1998. A review of machine scheduling: Complexity, algorithms and approximability. *Handbook of combinatorial optimization* **3** 21–169.
- Cho, Danny I., Mahmut Parlar. 1991. A survey of maintenance models for multi-unit systems. *European Journal of Operational Research* **51**(1) 1–23.
- Dekker, Rommert, Ralph E. Wildeman, Frank A. van der Duyn Schouten. 1997. A review of multi-component maintenance models with economic dependence. *Mathematical Methods of Operations Research* **45**(3) 411–435.
- Frangopol, Dan M., Maarten-Jan Kallen, Jan M. van Noortwijk. 2004. Probabilistic models for life-cycle performance of deteriorating structures: review and future directions. *Progress in Structural Engineering and Materials* **6**(4) 197212.
- Guo, R., H. Ascher, C. E. Love. 2000. Generalized models of repairable systemsA survey via stochastic processes formalism. *ORiON* **16**(2) 87128.
- Hall, Leslie A., Andreas S. Schulz, David B. Shmoys, Joel Wein. 1997. Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. *Mathematics of Operations Research* **22**(3) 513–544.
- Hochbaum, Dorit S. 1997. Approximation algorithms for NP-hard problems. *ACM SIGACT News* **28**(2) 40–52.
- Hochbaum, Dorit S., David B. Shmoys. 1987. Using dual approximation algorithms for scheduling problems theoretical and practical results. *Journal of the ACM (JACM)* **34**(1) 144–162.
- Jackson, James R. 1955. Scheduling a production line to minimize maximum tardiness. Tech. rep., DTIC Document.
- Jardine, Andrew K. S., Daming Lin, Dragan Banjevic. 2006. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing* **20**(7) 1483–1510.
- Lawler, E. L. 1982. A fully polynomial approximation scheme for the total tardiness problem. *Operations Research Letters* **1**(6) 207–208.
- Lawler, Eugene L. 1973. Optimal sequencing of a single machine subject to precedence constraints. *Management Science* **19**(5) 544–546.
- Lawler, Eugene L., Jan Karel Lenstra, Alexander HG Rinnooy Kan, David B. Shmoys. 1993. Sequencing and scheduling: Algorithms and complexity. *Handbooks in operations research and management science* **4** 445–522.

- Lenstra, Jan Karel, AHG Rinnooy Kan, Peter Brucker. 1977. Complexity of machine scheduling problems .
- Leung, Joseph YT. 2004. *Handbook of scheduling: algorithms, models, and performance analysis*. CRC Press.
- McCall, John J. 1965. Maintenance policies for stochastically failing equipment: A survey. *Management Science* **11**(5) 493–524.
- McMahon, Graham, Michael Florian. 1975. On scheduling with ready times and due dates to minimize maximum lateness. *Operations Research* **23**(3) 475.
- McNaughton, Robert. 1959. Scheduling with deadlines and loss functions. *Management Science* **6**(1) 1–12.
- Mohring, Rolf H., Andreas S. Schulz, Marc Uetz. 1999. Approximation in stochastic scheduling: the power of LP-based priority policies. *Journal of the ACM* **46**(6) 924–942.
- Moore, J. Michael. 1968. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science* **15**(1) 102–109.
- Nicolai, Robin P., Rommert Dekker. 2008. *Optimal maintenance of multi-component systems: a review*. Springer.
- Pierskalla, William P., John A. Voelker. 1976. A survey of maintenance models: The control and surveillance of deteriorating systems. *Naval Research Logistics Quarterly* **23**(3) 353–388.
- Pinedo, Michael. 1983. Stochastic scheduling with release dates and due dates. *Operations Research* **31**(3) 559–572.
- Pinedo, Michael. 2012. *Scheduling: theory, algorithms, and systems*. Springer.
- Queyranne, Maurice, Andreas S. Schulz, Maurice Queyranne. 1994. *Polyhedral approaches to machine scheduling*. Citeseer.
- Rothkopf, Michael H. 1966. Scheduling with random service times. *Management Science* **12**(9) 707–713.
- Sherif, Y. S., M. L. Smith. 1981. Optimal maintenance models for systems subject to failure A review. *Naval Research Logistics Quarterly* **28**(1) 477–4.
- Sherif, Yosef S. 1982. Reliability analysis : Optimal inspection and maintenance schedules of failing systems. *Microelectronics Reliability* **22**(1) 59–115.
- Smith, Wayne E. 1956. Various optimizers for single-stage production. *Naval Research Logistics Quarterly* **3**(1-2) 59–66.
- Sun, Zhankun, Nilay Tank Argon, Serhan Ziya. 2014. Priority scheduling of jobs with unknown types .
- Valdez-Flores, Ciriaco, Richard M. Feldman. 1989. A survey of preventive maintenance models for stochastically deteriorating single-unit systems. *Naval Research Logistics (NRL)* **36**(4) 419–446.
- van Noortwijk, J. M. 2009. A survey of the application of gamma processes in maintenance. *Reliability Engineering & System Safety* **94**(1) 2–21.

- Wang, Hongzhou. 2002. A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research* **139**(3) 469–489.
- Weiss, Gideon, Michael Pinedo. 1980. Scheduling tasks with exponential service times on non-identical processors to minimize various cost functions. *Journal of Applied Probability* 187–202.
- Wierman, Adam, Misja Nuyens. 2008. Scheduling despite inexact job-size information. *ACM SIGMETRICS Performance Evaluation Review*, vol. 36. 25–36.
- Yamayee, Zia A. 1982. Maintenance scheduling: Description, literature survey, and interface with overall operations scheduling. *IEEE Transactions on Power Apparatus and Systems* **PAS-101**(8) 2770–2779.



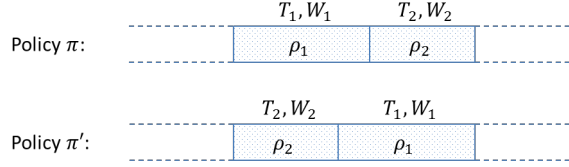


Figure 10 Example of processing jobs in decreasing order of ratio.

## Appendix A: Appendix

### A.1. Proof of Lemma 1

*Proof.* We prove by contradiction. Let  $\pi$  denote the optimal policy, and assume that at some point jobs 1 and 2 are processed one after the other, and that the ratio of job 1 is greater than the ratio of job 2. We introduce a policy  $\pi'$  which is identical to policy  $\pi$  except for processing job 2 immediately before processing job 1 (see Figure 10). It is easy to see that the interchange only affects jobs 1 and 2, as the expected completion time of any other job remains the same.

We denote by  $T_i, W_i$  the random processing time and weight of job  $i$ . One of the jobs refers to a known job with ratio  $t_i/w_i$ , while the other may refer to either another known job, or to an unknown job with ratio  $\rho$ . The completion time of job 1 increased by  $T_2$ , and the completion time of job 2 decreased by  $T_1$ . Therefore, the objective decreased by  $\mathbb{E}[T_1W_2 - T_2W_1]$ . We then obtain the following:

$$\begin{aligned} \mathbb{E}[T_1W_2 - T_2W_1] &= \mathbb{E}[T_1]\mathbb{E}[W_2] - \mathbb{E}[T_2]\mathbb{E}[W_1] \\ &= \mathbb{E}[W_1]\mathbb{E}[W_2] \left( \frac{\mathbb{E}[T_1]}{\mathbb{E}[W_1]} - \frac{\mathbb{E}[T_2]}{\mathbb{E}[W_2]} \right) \\ &> 0, \end{aligned}$$

whereas we use the independence between the two jobs for the first equality, and the assumption that the ratio of job 1 is higher than that of job 2 to obtain the inequality. This contradicts the optimality of policy  $\pi$ . Q.E.D.

### A.2. Proof of Lemma 2

*Proof.* By contradiction, assume that at some point of time the optimal policy  $\pi$  processes a job  $i$ , s.t.  $\mathbb{E}[T_i]/\mathbb{E}[W_i] > \rho_1$  (job 1 is a known job). Job  $i$  can be either a known or an unknown job. The resulting schedule is illustrated in Figure 11. We denote by  $L$  the time interval between the processing of jobs  $i$  and 1 during which testing and processing of other jobs might take place. Let  $W_L$  represent the total weight of jobs processed in the time intervals  $L$ , and let  $T_L$  denote the total duration of the time interval  $L$ . Note that for any policy  $\pi$ ,  $T_L$  and  $W_L$  random variables (as a special case these could be constants). In what follows we ignore the part of the objective corresponding to the elapsed time, as this can be seen as a sunk cost independent of future actions.

Under policy  $\pi$ , the expected weighted completion time of jobs  $i$  and 1 are  $\mathbb{E}[T_iW_i]$ , and  $\mathbb{E}[(T_i + T_L + t_1)w_1]$ . If  $C_L$  denotes the expected weighted completion times of the jobs processed in the time intervals  $L$  under policy  $\pi$ , then the part of the objective value corresponding to jobs  $i$ , 1 and the jobs processed in  $L$ , is

$$C_\pi = \mathbb{E}[T_iW_i + C_L + (T_i + T_L + t_1)w_1]. \quad (10)$$

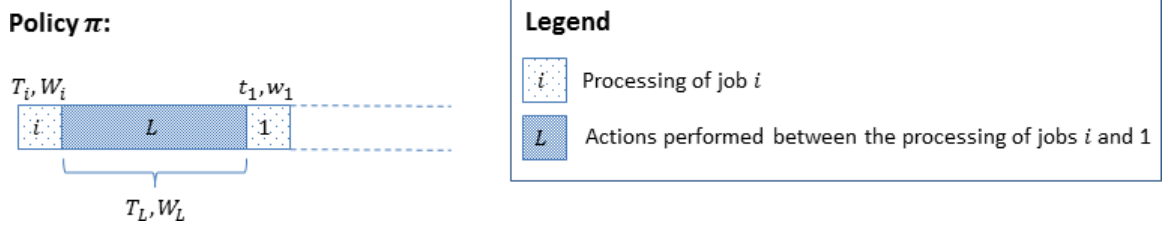


Figure 11 The schedule under policy  $\pi$

We now show that we can construct a policy  $\pi'$  that achieves a lower expected objective value. Policy  $\pi'$  imitates policy  $\pi$  over the interval  $L$ , while changing the order in which the processing of jobs 1 and  $i$ , and the actions in the interval  $L$  are performed.

We start with the cases when  $\mathbb{E}[W_L] = 0$ , that is, when no processing takes place in  $L$ . If  $\mathbb{E}[T_L] = 0$ , then the interval  $L$  is empty, and jobs  $i$  and 1 are processed consecutively. From Lemma 1 we know that policy  $\pi$  is not optimal. If  $\mathbb{E}[W_L] = 0$  and  $\mathbb{E}[T_L] > 0$ , then the policy  $\pi$  always performs tests (and only tests) in  $L$ . In this case  $C_L = 0$  (but  $T_L > 0$ ), and so  $C_\pi = \mathbb{E}[T_i W_i + (T_i + T_L + t_1) w_1]$ . An improved policy  $\pi'$  is identical to policy  $\pi$  except for processing job 1 before performing the test actions. It is easy to see that only job 1 is affected by this change and that its completion time decreases.

Assume now that under policy  $\pi$ ,  $\mathbb{E}[W_L] > 0$ . We define the ratio  $\rho_L = \mathbb{E}[T_L] / \mathbb{E}[W_L]$ . We are in one of three cases:

- (1)  $\rho_1 < \mathbb{E}[T_i] / \mathbb{E}[W_i] \leq \rho_L$ ;
- (2)  $\rho_1 < \rho_L < \mathbb{E}[T_i] / \mathbb{E}[W_i]$ ;
- (3)  $\rho_L \leq \rho_1 < \mathbb{E}[T_i] / \mathbb{E}[W_i]$ .

Note that we can separate to different cases due to the independence of  $W_i$  and  $T_i$ . For any policy  $\pi$ , the ratio  $\rho_L$  is fixed and can be determined in advance before taking any action. Calculating  $\rho_L$  might be complicated and computationally challenging, but nonetheless, can be done in finite time.

Case 1,  $\rho_1 < \mathbb{E}[T_i] / \mathbb{E}[W_i] \leq \rho_L$ . We construct the policy  $\pi'$  as follows: first process job 1, process job  $i$ , and then perform the actions in the interval  $L$ . The relevant part of the objective value under policy  $\pi'$  is

$$C_{\pi'} = \mathbb{E}[t_1 w_1 + (t_1 + T_i) W_i + (C_L + t_1 W_L)],$$

whereas the weighted time of job 1 is  $t_1 w_1$ , the weighted time of job  $i$  is  $(t_1 + T_i) W_i$ , and the weighted times of the jobs in interval  $L$  is  $C_L + t_1 W_L$ . The latter is due to a delay of length  $t_1$  to the jobs in  $L$ , which increased the objective value by  $t_1 W_L$ . The difference in the objective values between the two policies is therefore

$$\begin{aligned} C_\pi - C_{\pi'} &= \mathbb{E}[T_i W_i + C_L + (T_i + T_L + t_1) w_1] - \mathbb{E}[t_1 w_1 + (t_1 + T_i) W_i + (C_L + t_1 W_L)] \\ &= \mathbb{E}[T_i w_1 + T_L w_1 - t_1 W_i - t_1 W_L] \\ &= \mathbb{E}[T_i w_1 - t_1 W_i] + \mathbb{E}[T_L w_1 - t_1 W_L] \\ &= (\mathbb{E}[T_i] w_1 - t_1 \mathbb{E}[W_i]) + (\mathbb{E}[T_L] w_1 - t_1 \mathbb{E}[W_L]) \\ &= w_1 \mathbb{E}[W_i] \left( \frac{\mathbb{E}[T_i]}{\mathbb{E}[W_i]} - \frac{t_1}{w_1} \right) + w_1 \mathbb{E}[W_L] \left( \frac{\mathbb{E}[T_L]}{\mathbb{E}[W_L]} - \frac{t_1}{w_1} \right). \end{aligned}$$

Since  $\rho_1 < \mathbb{E}[T_i]/\mathbb{E}[W_i] \leq \rho_L$ , we obtain that  $C_\pi - C_{\pi'} > 0$  and that policy  $\pi'$  is sub-optimal.

Similarly to Case 1, one can construct policies that outperform policy  $\pi$  under the assumptions of Cases 2 and 3. For Case 2, policy  $\pi'$  is similar to policy  $\pi$  with an interchange between the processing of jobs  $i$  and 1. For Case 3, in policy  $\pi'$  we first process the jobs in  $L$ , then we process jobs 1, and finally we process job  $i$  and imitate policy  $\pi$  thereafter. This contradicts the assumption that processing a job  $i$  when  $\rho_1 < \mathbb{E}[T_i]/\mathbb{E}[W_i]$  is optimal. Q.E.D.

### A.3. Proof of Lemma 4

*Proof.* Using Lemma 2, we know that processing any other job is sub-optimal. It is left to show that testing is also sub-optimal. We prove this by induction on the number of unknown jobs  $N$ .

*Base,  $N = 1$ :* Assume by contradiction that the under the optimal policy  $\pi$ , the control at state  $(N, [t_1, w_1, \dots, t_n, w_n])$  is to test. We construct a policy  $\pi'$ , which processes job 1, tests the unknown job, and then follows policy  $\pi$  (Figure 2). From Lemma 1, we know that after testing, both policies process all jobs according to the WSPT rule. The difference in the objective value between the two policies is due to (1) an additional testing delay under policy  $\pi$ , and (2) sub-optimal processing order under policy  $\pi'$  when the ratio of the tested job is smaller than  $\rho_1$ . We obtain the following:

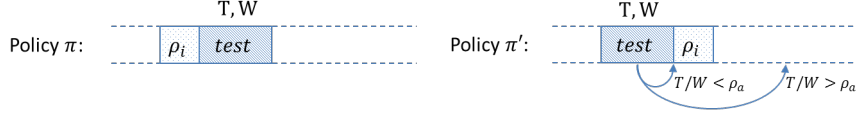
$$\begin{aligned} J_{mrg}^\pi(1, [t_1, w_1, \dots]) - J_{mrg}^{\pi'}(1, [t_1, w_1, \dots]) &= t_a w_1 + Prob\left(\frac{T}{W} < \frac{t_1}{w_1}\right) \mathbb{E}\left[T w_1 - t_1 W \mid \frac{T}{W} < \frac{t_1}{w_1}\right] \\ &= t_a w_1 - Prob\left(\frac{T}{W} < \frac{t_1}{w_1}\right) \mathbb{E}\left[t_1 W - T w_1 \mid \frac{T}{W} < \frac{t_1}{w_1}\right] \\ &= t_a w_1 - \mathbb{E}\left[(t_1 W - T w_1)^+\right] \\ &= w_1 \left(t_a - \mathbb{E}\left[(\rho_1 W - T)^+\right]\right) > 0, \end{aligned}$$

which contradict the optimality of policy  $\pi$ .

*Step,  $N > 1$ :* By contradiction, assume that the optimal control under policy  $\pi$  is test. Using the induction hypothesis one of two scenarios materializes: (1) the ratio of the tested job is greater than  $\rho_1$  in which case job 1 is processed immediately after testing; (2) the ratio of the tested job is less than  $\rho_1$  in which case, the tested job is processed immediately, followed by the processing of job 1. We compare the objective of policy  $\pi$  with a policy  $\pi'$  that tests only after processing job 1 (see Figure 2). In both cases, the difference in the objective is an outcome of additional testing times (under policy  $\pi$  there is one more job in the system when testing takes place), and there are savings to policy  $\pi$  from an improved processing order of job 1 and the tested job. We obtain the following expression for the difference:

$$\begin{aligned} J_{mrg}^\pi(N, [t_1, w_1, \dots]) - J_{mrg}^{\pi'}(N, [t_1, w_1, \dots]) &= t_a w_1 + Prob\left(\frac{T}{W} < \frac{t_1}{w_1}\right) \mathbb{E}\left[-t_1 W + T w_1 \mid \frac{T}{W} < \frac{t_1}{w_1}\right] \\ &= t_a w_1 - \mathbb{E}\left[(t_1 W - T w_1)^+\right] \\ &= t_a w_1 - w_1 \mathbb{E}\left[\left(\frac{t_1}{w_1} W - T\right)^+\right] \\ &= w_1 \left(t_a - \mathbb{E}\left[(\rho_1 W - T)^+\right]\right) \\ &> 0. \end{aligned}$$

Since policy  $\pi'$  has a lower objective, we obtain a contradiction to the optimality of policy  $\pi$ , which proves that testing and processing unknown jobs before processing job 1 are not optimal. Q.E.D.



**Figure 12** Example: testing after processing a medium- or high-ratio job.

#### A.4. Proof of Lemma 5

*Proof.* By contradiction, assume that under the optimal policy  $\pi$  testing is performed immediately after processing job  $i$ , such that  $\rho_a < \rho_i$  (Figure 12). Compare policy  $\pi$  with a policy  $\pi'$  that tests before processing job  $i$ . If the ratio of the tested job is lower than  $\rho_a$  it is processed immediately (Lemma 4), otherwise, policy  $\pi'$  processes job  $i$  and imitates policy  $\pi$  thereafter. Using marginal cost accounting, we obtain the following difference in their objective:

$$\begin{aligned}
 J_{mrg}^{\pi'}(N, [t_1, w_1, \dots]) - J_{mrg}^{\pi}(N, [t_1, w_1, \dots]) &= \mathbb{E}_{T_i, W_i} \left[ t_a W_i - Prob \left( \frac{T}{W} < \rho_a \right) \mathbb{E}_{T, W} \left[ T_i W - T W_i \mid \frac{T}{W} < \rho_a \right] \right] \\
 &= t_a \mathbb{E}[W_i] - Prob \left( \frac{T}{W} < \rho_a \right) \mathbb{E}_{T, W, T_i, W_i} \left[ T_i W - T W_i \mid \frac{T}{W} < \rho_a \right] \\
 &= t_a \mathbb{E}[W_i] - Prob \left( \frac{T}{W} < \rho_a \right) \mathbb{E}_{T, W} \left[ \mathbb{E}[T_i] W - T \mathbb{E}[W_i] \mid \frac{T}{W} < \rho_a \right] \\
 &= \mathbb{E}[W_i] \left( t_a - Prob \left( \frac{T}{W} < \rho_a \right) \mathbb{E} \left[ \frac{\mathbb{E}[T_i]}{\mathbb{E}[W_i]} W - T \mid \frac{T}{W} < \rho_a \right] \right) \\
 &< \mathbb{E}[W_i] \left( t_a - Prob \left( \frac{T}{W} < \rho_a \right) \mathbb{E} \left[ \rho_a W - T \mid \frac{T}{W} < \rho_a \right] \right) \\
 &= \mathbb{E}[W_i] \left( t_a - \mathbb{E} \left[ (\rho_a W - T)^+ \right] \right) \\
 &= 0.
 \end{aligned}$$

We obtain the first equality by comparing the objective of the two policies, conditioning on the ratio of the tested job. If the tested job has low-ratio, it is processed immediately after testing. The difference between the two policies is in the additional testing time, and the processing order of job  $i$  and the tested job. The difference in the objective is:  $Prob \left( \frac{T}{W} < \rho_a \right) \mathbb{E}_{T_i, W_i} [t_a W_i - \mathbb{E}_{T, W} [T_i W - T W_i \mid \frac{T}{W} < \rho_a]]$ ; otherwise, if the ratio of the tested job is not low, the processing order is the same under both policies, while there is still the additional testing cost of:  $Prob \left( \frac{T}{W} > \rho_a \right) \mathbb{E}_{T_i, W_i} [t_a W_i]$ . The third equality results from the independence of job  $i$  and the tested job, and the following inequality holds since  $\rho_a < \rho_i$ . Note that job  $i$  is allowed to have a random processing time and weight to cover the case when job  $i$  is an unknown job that is processed without testing. Q.E.D.

#### A.5. Proof of Theorem 2

*Proof.* We prove by induction on  $N$ , that for any state  $s = (N, [t_1, w_1, \dots, t_n, w_n])$ , testing is not optimal. Denote by *test* the policy that tests the unknown job at state  $s$ .

Starting with  $N = 1$ , the case with no known jobs is trivial. If  $\rho_1 < \rho$ , then Lemma 4 implies that policy *test* is not optimal and the claim holds. Otherwise, if  $\rho < \rho_1$ , we compare policy *test* with the the policy *proc-all*, which processes all jobs in non-decreasing order of their ratio (in which case the unknown job has the lowest ratio of all jobs). Using marginal cost accounting, the difference in the objective value between

the two policies results from the additional testing time under policy *test*, and the sub-optimal ordering of policy *proc – all*. The difference is therefore:

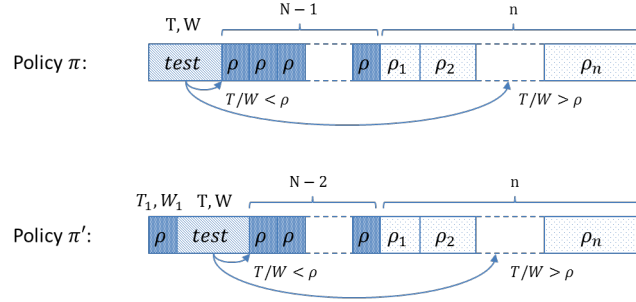
$$\begin{aligned}
J_{mrg}^{test}(1, [t_1, w_1, \dots]) - J_{mrg}^{proc-all}(1, [t_1, w_1, \dots]) &= t_a \sum w_i + \sum \mathbb{E}[\min(Wt_i, w_i T)] - \sum w_i \mathbb{E}[T] \\
&= t_a \sum w_i + \sum w_i \mathbb{E} \left[ \min \left( W \frac{t_i}{w_i}, T \right) \right] - \sum w_i \mathbb{E}[T] \\
&= \sum w_i (t_a - \mathbb{E}[T - \min(W\rho_i, T)]) \\
&= \sum w_i \left( \mathbb{E}[(\rho_a W - T)^+] - \mathbb{E}[(T - \rho W)^+] \right) \\
&> \sum w_i \left( \mathbb{E}[(\rho W - T)^+] - \mathbb{E}[(T - \rho W)^+] \right) \\
&= 0.
\end{aligned}$$

In the fourth equality we use Definition 1 to substitute for  $t_a$  with an equivalent expression. The last equality results from simple arithmetic. This shows that  $J_{mrg}^{test}(1, [t_1, w_1, \dots]) > J_{mrg}^{proc-all}(1, [t_1, w_1, \dots])$ , which means that testing is not optimal for the base case.

Now, assume that  $N > 1$  and the claim holds for  $N - 1$ . Once again, if  $\rho_1 < \rho$  the claim follows from Lemma 4. For  $\rho < \rho_1$ , the proof follows by contradiction. Assume that testing is optimal under the optimal policy  $\pi$  (see Figure 13). By the induction hypothesis, after testing we have  $N - 1$  unknown jobs and the optimal policy  $\pi$  processes all jobs in a non-decreasing order of their ratio. Thus, if the tested job's ratio is smaller than  $\rho$ , it will be processed before processing the next unknown job (there is at least one such job since  $N > 1$ ). If the tested job has a greater ratio, then it is processed at some point after processing the first unknown job. We compare the objective value of policy  $\pi$  with that of policy  $\pi'$  that first processes an unknown job, then tests, and then processes all jobs in a non-decreasing order of their ratio. The difference in the objective value between the two policies includes an ordering costs, and testing costs. We denote by  $T, W, T_1, W_1$  the processing times and the weights of the tested and the processed (unknown) job respectively. The difference is:

$$\begin{aligned}
J_{mrg}^{\pi}(N, [t_1, w_1, \dots]) - J_{mrg}^{\pi'}(N, [t_1, w_1, \dots]) &= \mathbb{E}_{T_1, W_1} \left[ t_a W_1 + Prob \left( \frac{T}{W} < \rho \right) \mathbb{E}_{T, W} \left[ TW_1 - WT_1 \mid \frac{T}{W} < \rho \right] \right] \\
&= t_a \mathbb{E}[W_1] - Prob \left( \frac{T}{W} < \rho \right) \mathbb{E}_{T_1, W_1, T, W} \left[ WT_1 - TW_1 \mid \frac{T}{W} < \rho \right] \\
&= t_a \mathbb{E}[W_1] - Prob \left( \frac{T}{W} < \rho \right) \mathbb{E}_{T, W} \left[ W \mathbb{E}[T_1] - T \mathbb{E}[W_1] \mid \frac{T}{W} < \rho \right] \\
&= t_a \mathbb{E}[W_1] - \mathbb{E}[W_1] Prob \left( \frac{T}{W} < \rho \right) \mathbb{E} \left[ W \frac{\mathbb{E}[T_1]}{\mathbb{E}[W_1]} - T \mid \frac{T}{W} < \rho \right] \\
&= t_a \mathbb{E}[W_1] - \mathbb{E}[W_1] Prob \left( \frac{T}{W} < \rho \right) \mathbb{E} \left[ W\rho - T \mid \frac{T}{W} < \rho \right] \\
&= \mathbb{E}[W_1] \left( t_a - \mathbb{E}[(W\rho - T)^+] \right) \\
&> 0.
\end{aligned}$$

The first equality is a direct outcome of marginal cost accounting. For the third equality we use the independence between the tested and processed unknown jobs. Finally, the inequality results from Definition 1, using the assumption that  $\rho < \rho_a$ . The objective value under policy  $\pi$  exceeds the one under policy  $\pi'$ , which is a contradiction to the optimality of testing. Q.E.D.



**Figure 13** Example: testing when  $\rho < \rho_a$ .

#### A.6. Proof of Theorem 4

*Proof.* There are two directions to be proved. If

$$\begin{aligned} \left( N\mathbb{E}[W] + \sum_i w_i \right) t_a &< (N-1)\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] \\ &+ \sum_{i \in \text{Medium}} \mathbb{E}[(Wt_i - w_iT)^+] + \sum_{i \in \text{High}} \mathbb{E}[(w_iT - Wt_i)^+], \end{aligned}$$

then it is straightforward from Lemma 9 that policy *STP* outperforms policy *PA*, and therefore by Theorem 1 the optimal control is test.

We prove the second direction by induction on  $N$ . For the base case  $N = 1$ , there are only two possibilities at state  $(N, [t_1, w_1, \dots, t_n, w_n])$ : (1) process the job with the smallest expected ratio, which implies processing all of the jobs (Theorem 1), or (2) test the unknown job, in which case there are no additional unknown jobs left, and all jobs will be processed after a single test was done. These correspond to the policies *PA* and *STP*. We obtain the condition of Equation (7) directly from Lemma 9.

We now assume that the assumption holds for any  $N - 1$ , and show that it holds for  $N$ . We first show that if

$$\begin{aligned} \left( N\mathbb{E}[W] + \sum_i w_i \right) t_a &\geq (N-1)\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] \\ &+ \sum_{i \in \text{Medium}} \mathbb{E}[(Wt_i - w_iT)^+] + \sum_{i \in \text{High}} \mathbb{E}[(w_iT - Wt_i)^+], \end{aligned}$$

then after testing a job, we reach a state that satisfies the same condition.

There are three cases for the realization of the tested job:

1. Low-ratio job
2. Medium-ratio job
3. High-ratio job

Case 1: the low-ratio job is immediately processed in which case the lefthand side (LHS) decreases by  $\mathbb{E}[W]t_a$ . The righthand side (RHS), decreases by  $\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+]$ . Since

$$\begin{aligned} t_a &= \mathbb{E}[(\rho_a W - T)^+] \\ &< \mathbb{E}[(\rho W - T)^+] \\ &= \frac{1}{\mathbb{E}[W]} \mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+], \end{aligned}$$

we obtain that  $\mathbb{E}[W]t_a < \mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+]$ , and LHS remains larger.

Case 2: In this case, LHS increases by  $(-\mathbb{E}[W] + w_i)t_a$ , and RHS increases by:

$$-\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] + \mathbb{E}[(Wt_i - w_iT)^+],$$

where  $\rho_i < \rho$ . A sufficient condition for  $\text{LHS} \geq \text{RHS}$  is

$$\begin{aligned} & (-\mathbb{E}[W] + w_i)t_a \geq -\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] + \mathbb{E}[(Wt_i - w_iT)^+] \\ \Leftrightarrow & (w_i)t_a - \mathbb{E}[(Wt_i - w_iT)^+] \geq (\mathbb{E}[W])t_a - \mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] \\ \Leftrightarrow & w_i(t_a - \mathbb{E}[(W\rho_i - T)^+]) \geq \mathbb{E}[W](t_a - \mathbb{E}[(W\rho - T)^+]) \\ \Leftrightarrow & w_i(\mathbb{E}[(W\rho_i - T)^+] - t_a) \leq \mathbb{E}[W](\mathbb{E}[(W\rho - T)^+] - t_a). \end{aligned}$$

We now look at the last inequality for the realization of the medium-ratio job that satisfy the inequality. The RHS does not depend on the realization. The LHS is increasing in  $t_i$ , and decreasing in  $w_i$ . This means that if a realization  $(t_i, w_i)$  satisfy the inequality, then the points  $(t_i, w_i + \epsilon)$ ,  $(t_i - \epsilon, w_i)$ , and  $(ct_i, cw_i)$  (where  $c < 1$ ) will also satisfy the inequality. The inequality holds as an equality when  $(t_i, w_i) = (\mathbb{E}[T], \mathbb{E}[W])$ , therefore it will also hold for all the points  $(t, w_i)$  such that  $\rho_a < t/w < \rho$  and  $t_i \leq \mathbb{E}[T]$ .

Case 3: As in case 2, LHS increases by  $(-\mathbb{E}[W] + w_i)t_a$ , and RHS increases by:

$$-\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] + \mathbb{E}[(w_iT - Wt_i)^+],$$

where  $\rho_i > \rho$ . A sufficient condition for  $\text{LHS} \geq \text{RHS}$  is

$$\begin{aligned} & (-\mathbb{E}[W] + w_i)t_a \geq -\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] + \mathbb{E}[(w_iT - Wt_i)^+] \\ \Leftrightarrow & (w_i)t_a - \mathbb{E}[(w_iT - Wt_i)^+] \geq (\mathbb{E}[W])t_a - \mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] \\ \Leftrightarrow & w_i(t_a - \mathbb{E}[(T - W\rho_i)^+]) \geq \mathbb{E}[W](t_a - \mathbb{E}[(W\rho - T)^+]) \\ \Leftrightarrow & w_i(t_a - \mathbb{E}[(T - W\rho_i)^+]) \geq \mathbb{E}[W](t_a - \mathbb{E}[(T - W\rho)^+]) \\ \Leftrightarrow & w_i(\mathbb{E}[(T - W\rho_i)^+] - t_a) \leq \mathbb{E}[W](\mathbb{E}[(T - W\rho)^+] - t_a) \\ \Leftrightarrow & w_i(\mathbb{E}[(T - W\rho_i)^+] - t_a) \leq \mathbb{E}[W](\mathbb{E}[(W\rho - T)^+] - t_a). \end{aligned}$$

Similarly to Case 2, we want to find the range of values  $(t_i, w_i)$  for which the inequality holds. Observe that the RHS is always positive, and does not depend on the realization of the high-ratio job. The LHS is monotonically increasing in  $w_i$ , and monotonically decreasing in  $t_i$ . This means that if a point  $(t_i, w_i)$  satisfies the inequality, then so will the points  $(t_i, w_i - \epsilon)$ ,  $(t_i + \epsilon, w_i)$ ,  $(ct_i, cw_i)$  where  $c < 1$ . Since the point  $(t_i, w_i) = (\mathbb{E}[T], \mathbb{E}[W])$  satisfy the inequality (with equality), we deduce that all the high-ratio jobs with  $w_i \leq \mathbb{E}[W]$  also satisfy the inequality.

By the induction hypothesis, when the condition of Equation (7) is met, a policy that tests a job necessarily processes all jobs in the next state, and therefore is the *STP* policy. Lemma 9 guarantees that if

$$\begin{aligned} & \left( N\mathbb{E}[W] + \sum_i w_i \right) t_a \geq (N-1)\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] \\ & \quad + \sum_{i \in \text{Medium}} \mathbb{E}[(Wt_i - w_iT)^+] + \sum_{i \in \text{High}} \mathbb{E}[(w_iT - Wt_i)^+], \end{aligned}$$

then policy *PA* is better than policy *STP*, which means that processing all jobs is the optimal control. Q.E.D.

State	$J_{PA}$	$J_{mrg}$	Optimal control
(0, [3, 100], [1, 110])	0	0	process all
(0, [100, 3], [110, 1])	0	0	process all
(0, [100, 100], [110, 110])	0	0	process all
(0, [100], [110])	0	0	process all
(1, [3], [1])	115.96	115.96	process all
(2, [], [])	235.11	234.93	test one
(0, [3, 3], [1, 1])	0	0	process all
(1, [], [])	112.97	112.97	process all
(0, [], [])	0	0	
(0, [3], [1])	0	0	process all
(1, [100], [110])	419.97	386.79	test one

Figure 14 The solution to the model.

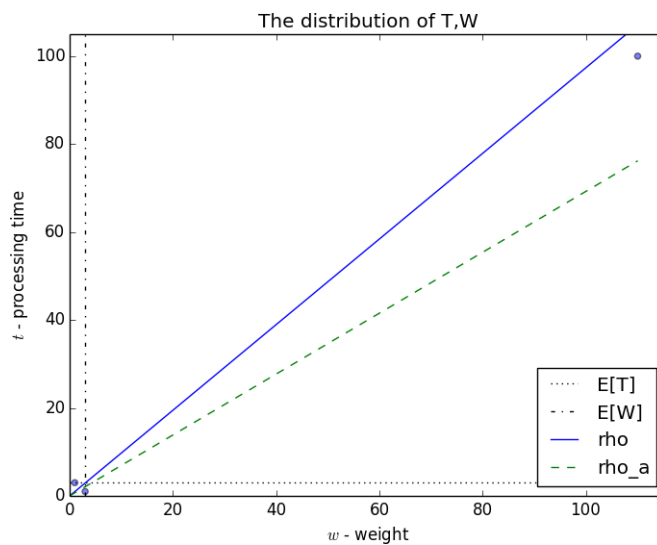


Figure 15 The Distribution  $\mathcal{D}_{T,W}$

### A.7. Example for a model where the myopic policy that is not optimal

Consider the following model:  $N_0 = 2$ ,  $t_a = 0.53$ , and where the distribution  $\mathcal{D}_{T,W}$  is given by:  $P(3, 1) = 0.5$ ,  $P(1, 3) = 0.49$ ,  $P(100, 110) = 0.01$ . The parameters  $\rho$  and  $\rho_a$  are equal to 0.97, and 0.69, respectively (see Figure 15 for an illustration).

The dynamic programming solution to the model is summarized in Table 14. For every state, it shows the value function of the optimal policy ( $J_{mrg}$ ), the cost of processing all jobs ( $J_{PA}$ ), and the optimal controls. While the myopic policy chooses to process all jobs at the initial state  $(2, [], [])$ , the optimal policy performs a test, and based on the realization, either processes all jobs or performs an additional test.

### A.8. Proof of Theorem 5

*Proof.* From Lemma 14, we know that it is optimal to test as long as the following holds:

$$\beta > \frac{N + \omega_H / \mathbb{E}[W]}{N - 1}.$$

In the initial state of the system,  $N = N_0$  and  $n = 0$ . Each time we test, the denominator  $(N - 1)$  decreases by 1, and the nominator  $(N + \omega_H / \mathbb{E}[W])$  increases by at most 1 (since  $w_i \leq 2\mathbb{E}[W]$ ). In the worst case, after



$N_{tests}$  tests (and processing of low-ratio jobs), the righthand side is at most  $(N_0 + N_{tests}) / (N_0 - 1 - N_{tests})$ . A sufficient condition for testing after  $N_{tests}$  periods is

$$\beta > \frac{(N_0 + N_{tests})}{(N_0 - 1 - N_{tests})},$$

which implies that the control *test one* is optimal for at least

$$N_{tests} = \left\lfloor N_0 \frac{(\beta - 1)}{(\beta + 1)} - \frac{\beta}{\beta + 1} \right\rfloor$$

periods. Q.E.D.

### A.9. Proof of Lemma 6

*Proof.* By induction on  $N$ . Base: for  $N = 0$  the claim holds as  $J_{LD}(N, \omega_M, \omega_H, \tau_M, \omega\tau) = 0$ .

Now assume the claim holds true for some  $N - 1$ . The value function is the minimum of two expressions. Since  $\tau_M$  and  $\omega\tau$  are non-negative, the first (test-one) is non-decreasing in  $\tau_M$  and  $\omega\tau$ , and  $J_{LD}(N - 1, \cdot, \cdot, \cdot, \cdot)$  is also non-decreasing in  $\tau_M$  and  $\omega\tau$  by the induction hypothesis. The second term (process-all) is also non-decreasing in  $\tau_M$  and  $\omega\tau$ . Q.E.D.

### A.10. Proof of Lemma 7

*Proof.* By induction on  $N$ . Base:  $N = 1$ . Recall that:

$$J_{LD}(N, \omega_M, \omega_H, \tau_M, \omega\tau) = \min \begin{cases} \mathbb{E}[TW] + (\omega_M + \omega_H + \mathbb{E}[W])t_a + \omega\tau & \text{test one} \\ \mathbb{E}[TW] + \mathbb{E}[W]\tau_M + \mathbb{E}[T]\omega_H & \text{process all.} \end{cases} \quad (11)$$

The value function under the control test-one is increasing in  $\omega\tau$ , and the value function under process-all is unaffected by changes in  $\omega\tau$ . This is the form of a threshold policy.

Step: once again, the value function under the control *process all* is unaffected by changes in  $\omega\tau$ , while the value function under the control *test one* is strictly increasing in  $\omega\tau$  (using Lemma 6). Q.E.D.

### A.11. Proof of Lemma 8

*Proof.* We want to show that for any state  $(N, \omega_M, \omega_H, \tau_M, \omega\tau)$ , the following holds:

$$1 \leq \frac{J_\delta(N, \omega_M, \omega_H, \tau_M, \omega\tau)}{J_{LD}(N, \omega_M, \omega_H, \tau_M, \omega\tau)} \leq (1 + \delta)^N.$$

We prove this by induction on  $N$ . We show that for each state and under each control, the ratio in the value functions between the two DP formulation is greater than or equal to 1, and smaller than or equal to  $(1 + \delta)^N$ .

Starting with the base case,  $N = 0$ , the control test-one is not feasible (no unknown jobs). Under the control process-all the costs are identical for both formulations and the lemma holds.

Before we continue to the step, we prove two properties about the ADP formulation.

LEMMA 15. *The value function  $J_\delta(N, \omega_M, \omega_H, \tau_M, \omega\tau)$  is non-decreasing in  $\omega_M, \omega_H, \tau_M, \omega\tau$ .*

*Proof.* The proof is similar to the proof of Lemma 6 and thus omitted.

LEMMA 16. *The following holds:*

$$\frac{J_\delta(N, \omega'_M, \omega'_H, \tau'_M, \omega\tau')}{J_\delta(N, \omega_M, \omega_H, \tau_M, \omega\tau)} \leq 1 + \delta,$$

where  $x \leq x' \leq (1 + \delta)x$  for  $x \in \{\omega_M, \omega_H, \tau_M, \omega\tau\}$ .

*Proof.* We prove by induction on  $N$ , starting with the base case  $N = 0$ . Testing is not allowed (no unknown jobs), therefore the value function is equal to the cost-to-go when choosing the control process-all:

$$\frac{J_\delta(N, \omega'_M, \omega'_H, \tau'_M, \omega\tau')}{J_\delta(N, \omega_M, \omega_H, \tau_M, \omega\tau)} = \frac{\mathbb{E}[W] N \tau'_M + N \mathbb{E}[T] \omega'_H + N \mathbb{E}[TW] + \binom{N}{2} \mathbb{E}[T] \mathbb{E}[W]}{\mathbb{E}[W] N \tau_M + N \mathbb{E}[T] \omega_H + N \mathbb{E}[TW] + \binom{N}{2} \mathbb{E}[T] \mathbb{E}[W]}.$$

This expression is bounded by  $1 + \delta$  as  $\tau_M$  and  $\omega_H$  satisfy:  $\tau'_M \leq (1 + \delta)\tau_M$ ,  $\omega'_H \leq (1 + \delta)\omega_H$ .

For the step, we assume the induction hypothesis holds for  $N - 1$  and prove that it holds for  $N$ . We show that for each of the controls the ratio between the value functions under the control is bounded by  $1 + \delta$ .

For the control process-all the ratio is bounded similarly to the base case. For the control test-one we observe that the cost function part satisfies:

$$\mathbb{E}[TW] + (\omega'_M + \omega'_H + N \mathbb{E}[W]) t_a + \omega\tau' \leq (1 + \delta) (\mathbb{E}[TW] + (\omega_M + \omega_H + N \mathbb{E}[W]) t_a + \omega\tau),$$

and that the terms  $d(N - 1) \mathbb{E}[W]$  and  $J_\delta(N - 1, \lceil \omega_M \rceil, \lceil \omega_H \rceil, \lceil \tau_M \rceil, \lceil \omega\tau \rceil)$  are the same under both formulations. We are left to show that the following two inequalities hold:

$$J_\delta\left(\frac{N - 1, \lceil \omega'_M + v \rceil, \lceil \omega'_H \rceil, \lceil \tau'_M + d \rceil}{\lceil \omega\tau' + \mathbb{E}[\min(Tv, dW)]}\right) \leq (1 + \delta) J_\delta\left(\frac{N - 1, \lceil \omega_M + v \rceil, \lceil \omega_H \rceil, \lceil \tau_M + d \rceil}{\lceil \omega\tau + \mathbb{E}[\min(Tv, dW)]}\right),$$

and

$$J_\delta\left(\frac{N - 1, \lceil \omega'_M \rceil, \lceil \omega'_H + v \rceil, \lceil \tau'_M \rceil}{\lceil \omega\tau' + \mathbb{E}[\min(Tv, dW)]}\right) \leq (1 + \delta) J_\delta\left(\frac{N - 1, \lceil \omega_M \rceil, \lceil \omega_H + v \rceil, \lceil \tau_M \rceil}{\lceil \omega\tau + \mathbb{E}[\min(Tv, dW)]}\right).$$

These hold by the induction hypothesis, and using the following inequality which holds for  $v \geq 0$ :

$$\frac{\lceil x' + v \rceil}{\lceil x + v \rceil} \leq \frac{\lceil \lceil x \rceil + v \rceil}{\lceil x + v \rceil} \leq 1 + \delta.$$

The latter holds since that

$$\frac{\lceil x \rceil + v}{x + v} \leq \frac{\lceil x \rceil}{x} \leq 1 + \delta,$$

which implies that

$$\frac{\lceil \lceil x \rceil + v \rceil}{\lceil x + v \rceil} \leq 1 + \delta.$$

Q.E.D.

We are now ready to prove the step of the main lemma. Once again, we compare the value functions under each of the controls. For the control process-all the two value functions are identical by definition.

For the test-one control, we look at the terms composing the cost-to-go function:

$$\mathbb{E}[TW] + (\omega_M + \omega_H + N \mathbb{E}[W]) t_a + \omega\tau,$$

and,

$$d(N - 1) \mathbb{E}[W]$$

are identical in both formulations.

Using Lemma 16 and the induction hypothesis:

$$\begin{aligned} J_\delta(N - 1, \lceil \omega_M \rceil, \lceil \omega_H \rceil, \lceil \tau_M \rceil, \lceil \omega\tau \rceil) &\leq (1 + \delta) J_\delta(N - 1, \omega_M, \omega_H, \tau_M, \omega\tau) \\ &= (1 + \delta)(1 + \delta)^{N-1} J_{LD}(N - 1, \omega_M, \omega_H, \tau_M, \omega\tau) \\ &= (1 + \delta)^N J_{LD}(N - 1, \omega_M, \omega_H, \tau_M, \omega\tau). \end{aligned}$$

Similarly,

$$\begin{aligned} J_\delta \left( \begin{array}{l} N-1, \lceil \omega_M + v \rceil, \lceil \omega_H \rceil, \lceil \tau_M + d \rceil, \\ \lceil \omega\tau + \mathbb{E}[\min(Tv, dW)] \rceil \end{array} \right) &\leq (1+\delta) J_\delta \left( \begin{array}{l} N-1, \omega_M + v, \omega_H, \tau_M + d, \\ \omega\tau + \mathbb{E}[\min(Tv, dW)] \end{array} \right) \\ &\leq (1+\delta)(1+\delta)^{N-1} J_{LD} \left( \begin{array}{l} N-1, \omega_M + v, \omega_H, \tau_M + d, \\ \omega\tau + \mathbb{E}[\min(Tv, dW)] \end{array} \right) \\ &= (1+\delta)^N J_{LD} \left( \begin{array}{l} N-1, \omega_M + v, \omega_H, \tau_M + d, \\ \omega\tau + \mathbb{E}[\min(Tv, dW)] \end{array} \right) \end{aligned}$$

and,

$$\begin{aligned} J_\delta \left( \begin{array}{l} N-1, \lceil \omega_M \rceil, \lceil \omega_H + v \rceil, \lceil \tau_M \rceil, \\ \lceil \omega\tau + \mathbb{E}[\min(Tv, dW)] \rceil \end{array} \right) &\leq (1+\delta) J_\delta \left( \begin{array}{l} N-1, \omega_M, \omega_H + v, \tau_M, \\ \omega\tau + \mathbb{E}[\min(Tv, dW)] \end{array} \right) \\ &\leq (1+\delta)(1+\delta)^{N-1} J_{LD} \left( \begin{array}{l} N-1, \omega_M, \omega_H + v, \tau_M, \\ \omega\tau + \mathbb{E}[\min(Tv, dW)] \end{array} \right) \\ &= (1+\delta)^N J_{LD} \left( \begin{array}{l} N-1, \omega_M, \omega_H + v, \tau_M, \\ \omega\tau + \mathbb{E}[\min(Tv, dW)] \end{array} \right), \end{aligned}$$

which implies that

$$\frac{J_\delta(N, \omega_M, \omega_H, \tau_M, \omega\tau)}{J_{LD}(N, \omega_M, \omega_H, \tau_M, \omega\tau)} \leq (1+\delta)^N.$$

Using Lemma 15 and the fact that rounding only increases the parameters, it is easy to see that the lower bound on the above ratio holds as well. Q.E.D.

### A.12. Proof of Lemma 9

*Proof.* Using marginal cost accounting, testing an unknown job carries an additional expected weighted time of  $(N\mathbb{E}[W] + \sum_i w_i) t_a$ , but, in return, it improves the scheduling order. For any medium-ratio job, the ordering cost of the unknown job and a medium-ratio job  $i$ , is by default  $Wt_i$  (since the medium-ratio job comes before the unknown job). The savings occur when the realization of the tested job has a lower ratio, in which case the improved ordering cost is  $w_i T$ . The expected savings of the pair is therefore  $\mathbb{E}[(Wt_i - w_i T)^+]$ . Similarly, the expected savings of high-ratio jobs is  $\mathbb{E}[(w_i T - Wt_i)^+]$ , and the savings with respect to other unknown jobs is equal to  $\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+]$  (and also to  $\mathbb{E}[(\mathbb{E}[W]T - W\mathbb{E}[T])^+]$ ). Q.E.D.

### A.13. Proof of Lemmas 10, 11, and 12

We look at the expression  $\mathbb{E}[\min(T_1, T_2)]$ . Let  $Z$  be the r.v. defined as  $Z = \min(T_1, T_2)$ . Then,  $Pr(Z \leq z) = 1 - (1 - F(z))^2$ , and  $f_Z(z) = 2(1 - F(z))f(z)$ , which means that  $\mathbb{E}[Z] = \int 2(1 - F(z))f(z)zdz$ . We get the following equivalent bound on the “process all” policy:

$$\frac{J^{PA}(N, \square)}{J^{OPT}(N, \square)} \leq \frac{\mu}{\mathbb{E}[Z]} = \frac{\mu}{\int 2(1 - F(z))f(z)zdz}.$$

For uniform distribution:

$$\begin{aligned} \mathbb{E}[Z] &= \int_a^b 2(1 - F(z))f(z)zdz \\ &= \int_a^b 2 \left(1 - \frac{z-a}{b-a}\right) \frac{1}{b-a} z dz \\ &= \frac{2}{(b-a)^2} \left. \frac{bz^2}{2} - \frac{z^3}{3} \right|_a^b \\ &= \frac{1}{3(b-a)^2} (b+2a)(b-a)^2, \end{aligned}$$

which implies an approximation of

$$APX = \frac{(a+b)}{2} \frac{3(b-a)^2}{(b+2a)(b-a)^2} = \frac{3(b+a)}{2(b+2a)},$$

this last is monotonically decreasing in  $a$ .

For exponential distribution:

$$\begin{aligned} \mathbb{E}[Z] &= \int_0^\infty 2(1-F(z))f(z)zdz \\ &= \int_0^\infty 2(1-1+e^{-\lambda z})\lambda e^{-\lambda z}zdz \\ &= \int_0^\infty 2\lambda e^{-2\lambda z}zdz \\ &= \frac{1}{2\lambda}, \end{aligned}$$

which implies a 2-approximation.

$$\text{For } T \sim \begin{cases} a & p \\ b & 1-p \end{cases} :$$

$$\begin{aligned} \frac{\mu}{\mathbb{E}[Z]} &= \frac{ap+b(1-p)}{b(1-p)^2+a(1-(1-p)^2)} \\ &= \frac{ap+b(1-p)}{a+(b-a)(1-p)^2}, \end{aligned}$$

By plugging in  $a=0, p=1/2$ , we get a  $1/(1-p)=2$  approximation. For  $a=0, b=M, p=1-1/M$ , we get an  $M$  approximation.

#### A.14. Proof of Lemma 14

*Proof.* A sufficient condition for the optimality of testing is  $J^{STP} - J^{PA} < 0$ . Using Lemma 9, this is equivalent to:

$$\begin{aligned} \left( N\mathbb{E}[W] + \sum_i w_i \right) t_a &\leq (N-1)\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] \\ &\quad + \sum_{i \in \text{Medium}} \mathbb{E}[(Wt_i - w_iT)^+] \\ &\quad + \sum_{i \in \text{High}} \mathbb{E}[(w_iT - Wt_i)^+], \end{aligned}$$

and to

$$\begin{aligned} \left( N\mathbb{E}[W] + \sum_{i \in \text{High}} w_i \right) t_a &\leq (N-1)\mathbb{E}[(W\mathbb{E}[T] - \mathbb{E}[W]T)^+] \\ &\quad + \sum_{i \in \text{Medium}} w_i (\mathbb{E}[(W\rho_i - T)^+] - t_a) \\ &\quad + \sum_{i \in \text{High}} w_i \mathbb{E}[(T - W\rho_i)^+]. \end{aligned}$$

By plugging  $\beta$ , we obtain

$$\begin{aligned} \left( N\mathbb{E}[W] + \sum_{i \in \text{High}} w_i \right) t_a &\leq (N-1)\beta t_a \mathbb{E}[W] \\ &\quad + \sum_{i \in \text{Medium}} w_i (\mathbb{E}[(W\rho_i - T)^+] - t_a) \\ &\quad + \sum_{i \in \text{High}} w_i \mathbb{E}[(T - W\rho_i)^+]. \end{aligned}$$

Since the last two summations are positive ( $\rho_i > \rho_a$  implies that  $\mathbb{E}[(W\rho_i - T)^+] > t_a$ ), the following condition is sufficient for the inequality to hold:

$$N\mathbb{E}[W] + \sum_{i \in \text{High}} w_i \leq (N - 1)\beta\mathbb{E}[W],$$

or equivalently:

$$\beta > \frac{N + \frac{\omega_H}{\mathbb{E}[W]}}{N - 1}.$$