Static Routing in Stochastic Scheduling: Performance Guarantees and Asymptotic Optimality

Santiago R. Balseiro[1], David B. Brown[2], and Chen Chen[2]

[1]Graduate School of Business, Columbia University
[2]Fuqua School of Business, Duke University
srb2155@columbia.edu, dbbrown@duke.edu, cc459@duke.edu

First version: July 21, 2016
This version: May 19, 2018

Abstract

We study the problem of scheduling a set of J jobs on M machines with stochastic job processing times when no preemptions are allowed and the weighted sum of expected completion time objective. Our model allows for "unrelated" machines: the distributions of processing times may vary across both jobs and machines. We study *static routing policies*, which assign (or "route") each job to a particular machine at the start of the problem and then sequence jobs on each machine according to the WSEPT (weighted shortest expected processing time) rule. We discuss how to obtain a good routing of jobs to machines by solving a convex quadratic optimization problem that has $J \times M$ variables and only depends on the job processing distributions through their expected values. Our main result is an additive performance bound on the sub-optimality of this static routing policy relative to an optimal adaptive, non-anticipative scheduling policy. This result implies that such static routing policies are asymptotically optimal as the number of jobs grows large. In the special case of "uniformly related" machines - that is, machines differ only in their speeds - we obtain a similar but slightly sharper result for a static routing policy that routes jobs to machines proportionally to machine speeds. We also study the impact that dependence in processing times across jobs can have on the sub-optimality of the static routing policy. The main novelty in our work is deriving lower bounds on the performance of an optimal adaptive, non-anticipative scheduling policy; we do this through the use of an information relaxation in which all processing times are revealed before scheduling jobs and a penalty that appropriately compensates for this additional information.

Subject classifications: Stochastic scheduling, unrelated machines, dynamic programming, information relaxation duality, asymptotic optimality.

1 Introduction

There is a rich and well-developed literature on scheduling problems in the operations research and computer science communities, and many varieties of scheduling problems arise in a broad range of applications. In this paper, we study the problem of non-preemptively scheduling a set of J jobs on a set of M unrelated machines when job processing times are stochastic. Each job j has a positive weight w_j and must be processed non-preemptively by one machine. Machines operate in parallel and at any given time a machine can process at most one job. Processing times for a job depend both on the job as well as the machine that processes the job, and the processing time of a job is not fully known until a job is completed. The objective is to minimize the weighted sum of expected completion times; following the notation of Graham et al. (1979), this problem is written as $R||\mathbb{E}[\sum_{j=1}^{J} w_j C_j]$.

Hoogeveen et al. (2001) study the deterministic version of this problem and show that no polynomial time approximation scheme exists if $\mathcal{P} \neq \mathcal{NP}$. In the stochastic case, the problem may be formulated as a stochastic dynamic program in which we minimize over non-anticipative policies, but the effort in computing an optimal policy scales exponentially in the number of jobs. Furthermore, optimal policies may be complex: for example, Uetz (2003) shows that deliberately idling machines can be optimal even when machines are identical. Given the difficulty of the problem, we typically can only consider suboptimal, heuristic policies. In this paper, we study a class of static policies and analyze their performance relative to an optimal non-anticipative policy, with an aim of showing that these policies are asymptotically optimal in the regime of many jobs.

When processing times are deterministic, the problem reduces to two sets of decisions: first, to which machines to assign or *route* jobs and, second, how to sequence jobs optimally on each machine. For deterministic scheduling on one machine, Smith (1956) showed that the optimal sequencing of jobs is in decreasing order of the ratio of weight to processing time. Rothkopf (1966) extended this result to the single machine stochastic scheduling problem $1||\mathbb{E}[\sum_{j=1}^{J} w_j C_j]$, showing that sequencing jobs in decreasing order of the ratio of weight to expected processing time is optimal. This sequencing is referred to as the WSEPT (Weighted Shortest Expected Processing Time) rule. Motivated by these insights, we consider *static routing policies* that at the start of the problem route each job to a given machine, then sequence the jobs routed to each machine according to WSEPT. To find an optimal static routing policy, the task then reduces to optimizing over the routing.

The objective of the resulting optimization problem is quadratic but not convex. For the deterministic version of this problem, Sethuraman and Squillante (1999) and Skutella (2001) independently proposed a convex relaxation to efficiently compute a routing and showed that the resulting solution performs within a factor of $\frac{3}{2}$ of optimal. Inspired by these papers, we also use convex relaxations to optimize over static routing policies for the stochastic problem. This leads to convex quadratic optimization problems with the same form as in Sethuraman and Squillante (1999) and Skutella (2001) with job processing times set to their expected values. Our main result is establishing an additive performance guarantee on the sub-optimality of these static routing policies. A corollary to this result is that, as the number of jobs grows large relative to the number of machines, these static routing policies are asymptotically optimal, provided job weights, expected processing times, and the coefficients of variation of processing times are uniformly bounded.

The main challenge is establishing a good lower bound on the performance of an optimal non-anticipative, adaptive scheduling policy. The result in this paper is qualitatively similar to an additive guarantee developed by Möhring et al. (1999) for the case when processing times are identical across machines (in particular, see Corollary 4.1 of that paper) but extended to the case with unrelated machines. The key technology used by Möhring et al. (1999) to develop lower bounds are polymatroid inequalities that describe the *performance space* of expected completion times achievable by feasible policies. This builds upon similar inequalities in Schulz (1996) and Hall et al. (1997) for deterministic scheduling on identical machines; a key step in establishing these inequalities is effectively an aggregation of jobs across machines, which appears to rely critically on job times being the same across machines. As we show in Section 5, bounds based on such relaxations of the performance space can be generalized to the special case of uniformly related machines (i.e., machines differ only in their speeds), but it is unclear how to generalize this approach to the more general case of unrelated machines.

In contrast, we derive lower bounds on the performance of an optimal scheduling policy by using a duality approach based on information relaxations. This approach involves relaxing some or all of the non-anticipativity constraints and imposing a penalty that punishes violations of the non-anticipativity constraints. The theory behind this approach is developed in Brown et al. (2010)

and builds upon earlier work on "martingale duality methods", independently developed by Rogers (2002) and Haugh and Kogan (2004) for pricing American options. A penalty is said to be *dual feasible* if it does not penalize any non-anticipative policy in expectation. Brown et al. (2010) establish weak and strong duality results: with any information relaxation and any dual feasible penalty, we obtain a bound on the performance of an optimal policy and there exists an "ideal" dual feasible penalty such that the bound is tight.

In this paper we focus exclusively on *perfect* information relaxations in which all job processing times are revealed before making any scheduling decisions. The perfect information problem then reduces to solving a deterministic scheduling problem for a given realization of job processing times. Absent a penalty, the resulting lower bound may be quite weak, especially when job processing times are highly uncertain. As a result, penalties are needed, and we develop dual feasible penalties that improve the lower bound. An attractive feature of this approach is that the problem reduces to the analysis of deterministic scheduling problems in every sample path, and we can leverage ideas from deterministic scheduling in this analysis - namely, the convex relaxations developed by Sethuraman and Squillante (1999) and Skutella (2001). This allows us to "close the loop" and relate the resulting lower bound on performance of an optimal adaptive, non-anticipative policy to the performance of the static routing policy.

The paper is organized as follows. Section 1.1 reviews some relevant literature. In Section 2 we formulate the problem, and in Section 3 we discuss static routing policies. Section 4 presents the main performance analysis results and an overview of the key steps of the proof. Section 5 studies the special case of uniformly related machines, and we show how to obtain a slightly stronger result using a very simple static routing policy that routes jobs proportionally to machine speeds. In Section 6 we discuss the impact that dependence in processing times on jobs can have on the sub-optimality of static routing policies. In Section 7 we demonstrate the performance of the static routing policies and the lower bounds we develop on randomly generated examples with up to 1,000 jobs. Section 8 concludes. The main proofs are in Appendix A; additional proofs and derivations are in (online) Appendices B, C, and D.

1.1 Literature Review

Our paper has connections to several streams of papers that analyze different forms of scheduling problems using different techniques.

Time-Indexed Linear Programs In time-indexed formulations, the planning horizon is divided into discrete time periods and binary variables are used to indicate whether a job is processed in a machine in a given time period. Linear programming (LP) relaxations of time-index formulations provide strong lower bounds, which have been extensively used to develop approximation algorithms for deterministic scheduling problems (see, e.g., Hall et al. 1997; Schulz and Skutella 2002). In recent, related work, Skutella et al. (2016) study static routing policies for stochastic scheduling on unrelated machines. The static routing policies they consider are based on a novel time-indexed LP relaxation. This time-indexed LP relaxation is powerful in that the approach allows for extensions such as release dates before which jobs cannot be processed. Moreover, Skutella et al. (2016) derive strong constant factor approximation results for versions of the problem both with and without release dates. However, the approach requires a discretization of the time dimension, a potentially large number of variables, and a full input of all cumulative distributions of job processing times along the discretization. In contrast, the static routing policies we consider, while more limited in scope (e.g., we do not handle release dates), are simple and intuitive, only require solving a convex quadratic problem with $J \times M$ variables, and only use the job weights and the expected values of job processing times as inputs. Moreover, our focus is on the asymptotic regime with many jobs and we develop an additive performance bound with that goal in mind. Given that the policies that we study appear to be simple - especially compared to an optimal adaptive, non-anticipative policy - and use little information about processing time distributions, it is perhaps surprising that these policies can perform well.

Fluid Relaxations for High-multiplicity Scheduling The high-multiplicity version of a scheduling problem assumes that jobs can be partitioned into a small number of types with all jobs being identical within a type. When the number of jobs within each type is large, it has been shown that many scheduling problems can be well approximated using fluid relaxations in which discrete jobs are replaced with a flow of continuous fluid. This approximation builds on the observation that, when every machine processes many jobs, the contribution of each job is infinitesimal and the sum of realized processing times tends to concentrate around its mean.

Bertsimas et al. (2003) study a deterministic job-shop scheduling problem with the objective of minimizing the total holding cost. They develop an approximation algorithm based on appropriately rounding an optimal solution of a fluid relaxation and show that this algorithm is within an additive amount from optimal and is asymptotically optimal when the number of jobs is large. Fluid models have been applied in a number of stochastic problems as well, such as multi-class queueing networks (e.g., Maglaras 2000 and Nazarathy and Weiss 2009) and stochastic processing networks (e.g., Dai and Lin 2005 and Dai and Lin 2008).

Through the use of a fluid relaxation, it may well be possible to show that simple heuristic policies, like the static routing policy considered in our paper, are asymptotically optimal when the number of jobs is large. This involves showing, using the functional law of large numbers, that the system behavior under an admissible policy converges almost surely to a solution of the fluid problem. Asymptotic optimality then follows because the proposed policy is optimal for the fluid problem. Typically, most papers restrict attention to work-conserving policies that are independent of the size of the problem (see, e.g., Theorem 4.1 in Dai 1995). Such policies are not necessarily optimal in our setting: as discussed, Uetz (2003) shows that deliberately idling machines can be optimal even when machines are identical. Additionally, the fluid problem typically does not provide a bound for the finite, discrete stochastic system, and only establishes optimality in the limit (an exception is Bertsimas et al. 2003, who provide an additive performance guarantee based on their fluid relaxation analysis for a deterministic job shop scheduling problem).

An advantage of the perfect information relaxation approach is that it provides an additive performance bound on the sub-optimality of the static routing policy relative to an optimal adaptive, non-anticipative scheduling policy (which might not be work-conserving). This bound holds for problems of all sizes and not only in the limit, which in turn provides an explicit rate of convergence to optimality.

Primal-dual Schemas and Dual-fitting Our approach is in spirit similar to the primal-dual schema and the closely related dual-fitting approach used to design approximation algorithms in many combinatorial and stochastic optimization problems (see, e.g., Bar-Yehuda and Even 1981, Agrawal et al. 1995, Jain and Vazirani 2001). In such an approach, one typically uses a candidate primal feasible solution to design a dual feasible solution to a relaxation (typically, an LP relaxation) of the original problem; the objective value of this dual feasible solution then provides a bound that ideally can be related to the performance of the candidate solution.

The idea of "dual-fitting" is closely related, and involves constructing a good dual feasible solution by properly scaling an initial candidate dual solution. Both Anand et al. (2012) and Gupta et al. (2017) use a dual-fitting approach to study variations of scheduling problems. Anand et al. (2012) consider several deterministic scheduling problems with different objectives and preemption, whereas Gupta et al. (2017) study an online version of the stochastic and non-preemptive scheduling problem on unrelated machines with a weighted sum of expected completion times objective in which jobs arrive over time. Applying a dual fitting approach to the time-indexed LP, both of these papers derive constant-factor guarantees for the heuristic policies that they study.

Our approach can be viewed as a primal-dual scheme in that we use our "primal feasible solution" (the static routing policy) to generate dual variables that play a key role in the performance bound. Specifically, to obtain a performance bound, we first consider the (penalized) perfect information relaxation, which relaxes the non-anticipativity constraints. This problem is a deterministic scheduling problem for every realization of job processing times. We then consider a convex relaxation of this problem, and further consider a Lagrangian dual of this relaxation that dualizes the constraints that require each job be assigned to exactly one machine. This procedure provides a lower bound for any dual feasible penalty and any value of the Lagrange multipliers. By properly designing the penalty and setting the Lagrange multipliers to be equal to those from the convex relaxation used to generate the static routing policy, we show that the optimal value of the penalized perfect information problem is "close" to the performance of our static routing policy in every sample path.

Information Relaxations The use of information relaxations to this point has largely been as a method for computing numerical bounds on specific large-scale problem instances; in contrast, here we use the approach as a proof technique to derive analytical bounds. This follows recent work by Balseiro and Brown (2016); we refer the reader to that paper for a more thorough literature review on information relaxations. Balseiro and Brown (2016) study stochastic scheduling on identical parallel machines (in addition to some other problems) and recover some of the results from Möhring et al. (1999). Specifically, Balseiro and Brown (2016) study the WSEPT priority policy by combining performance space relaxations and information relaxations; this proof technique does not extend to the unrelated machine case. In addition to studying a more general stochastic scheduling problem than Balseiro and Brown (2016), our static routing policy is different than the WSEPT priority policy they study, our penalty is quite different than the one considered there (namely, we have a routing component that is critical for our results - see Section 4.3 below), and our analysis of the information relaxations relies on quadratic programming and Lagrangian duality whereas their analysis relies on linear programming relaxations of the performance space.

Offline Analysis Another line of work studies the performance of heuristic policies relative to the performance of an optimal "offline" algorithm that knows all processing times in advance (i.e., what we call the perfect information relaxation without penalty; see, e.g., Scharbrodt et al. 2006; Souza and Steger 2006). For example, Souza and Steger (2006) show that the expected competitive ratio of the WSEPT list priority policy - given by the expected ratio of the performance of WSEPT to that of the perfect information relaxation - is at most 3 when job processing times are identical across machines and exponentially distributed. In general, no non-anticipative policy can achieve asymptotic optimality with respect to the (unpenalized) perfect information benchmark. Thus, as in Möhring et al. (1999) and Skutella et al. (2016) and other papers, we measure the performance of heuristic policies relative to the performance of an optimal adaptive, non-anticipative policy.

2 Problem Formulation

We consider the problem of scheduling a set of jobs on unrelated machines with the objective of minimizing the weighted sum of expected completion times when no preemptions are allowed. Job processing times are stochastic, and the realized processing times of each job are learned only after completion of the job. Each job must be processed by one machine. Machines may work in parallel but, at any given time, a machine can process at most one job. We let $\mathcal{J} = \{1, \ldots, J\}$ denote a set of jobs to be scheduled on a set of machines $\mathcal{M} = \{1, \ldots, M\}$. We denote the processing time of job $j \in \mathcal{J}$ on machine $m \in \mathcal{M}$ by p_{jm}, which is a random variable with positive expected value (denoted by $\mathbb{E}[p_{jm}]$) and finite variance (denoted by $\mathbb{V}\mathrm{ar}[p_{jm}]$). In Remark 4.1 we discuss how our results extend when each job can only be processed by a subset of machines. With the exception of Section 6, we assume processing times are independent across jobs. For our main results in Section 4, we make no assumptions about the dependence of processing times across machines for each job. Because preemptions are not allowed, jobs are independent, and each job is assigned to exactly one machine in every sample path, the dependence structure across machines for each job has no impact on the expected performance of any non-anticipative scheduling policy.

We let C_j denote the completion time of job $j \in \mathcal{J}$: that is, C_j denotes the sum of the time until the processing of j starts and the realized processing time of j. Each job has a positive weight w_j and the objective is to minimize $\mathbb{E}[\sum_{j\in\mathcal{J}} w_j C_j]$. We let Π denote the set of non-anticipative, adaptive policies. In this model time is continuous and decision epochs correspond to times when some machine is available; a policy $\pi \in \Pi$ determines the next job to be processed at each decision epoch based on the information available at that time. We let C_j^π denote the completion time of job j under policy $\pi \in \Pi$. We can write the problem as

$$V^* = \min_{\pi\in\Pi} \mathbb{E}\left[\sum_{j\in\mathcal{J}} w_j C_j^\pi\right],$$

where V^* denotes the performance of an optimal non-anticipative adaptive policy.

3 Static Routing Policies

An optimal policy will generally assign jobs to machines dynamically based on the realizations of processing times. In general, solving the associated stochastic dynamic program (DP) for an optimal policy requires tracking the state of every job (uncompleted, completed, or assigned to a particular machine) as well as the elapsed processing times for all jobs currently in process; we provide a description of the DP in Appendix C. Given the difficulty of finding an optimal policy even for problems with few jobs, we instead consider a *static routing policy* that, at the start of the problem, assigns each job $j \in \mathcal{J}$ to machine $m \in \mathcal{M}$ with probability x_{jm}, independently across jobs, and then sequences the jobs assigned to each machine $m \in \mathcal{M}$ in decreasing order of weight per *expected* processing time $r_{jm} \triangleq w_j/\mathbb{E}[p_{jm}]$. This sequencing is referred to as the WSEPT (Weighted

Shortest Expected Processing Time first) rule. As mentioned in the introduction, Rothkopf (1966) showed that the WSEPT rule minimizes the expected total weighted completion time on each single machine. Thus, for a given routing of jobs to machines, the WSEPT rule is the optimal sequencing.

We let $\mathcal{X} = \{\mathbf{x} = (x_{jm})_{j\in\mathcal{J}, m\in\mathcal{M}} \in \mathbb{R}_{+}^{J\times M} : \sum_{m\in\mathcal{M}} x_{jm} = 1,\ \forall j \in \mathcal{J}\}$ denote the set of all routing matrices and $V^{\mathrm{R}}(\mathbf{x})$ denote the expected performance of the static routing policy when jobs are routed according to routing matrix $\mathbf{x} \in \mathcal{X}$. For machine $m \in \mathcal{M}$, $\prec_m$ denotes the total order that sorts jobs by the WSEPT rule: that is, jobs $i, j \in \mathcal{J}$ satisfy that $i \prec_m j$ if $r_{im} > r_{jm}$ (if $r_{im} = r_{jm}$, we use the convention that $i \prec_m j$ if and only if $i < j$). Using linearity of expectations and the fact that jobs are routed independently, we can then write $V^{\mathrm{R}}(\mathbf{x})$ as

$$V^{\mathrm{R}}(\mathbf{x}) = \sum_{j\in\mathcal{J}} w_j \sum_{m\in\mathcal{M}} x_{jm}\Big(\mathbb{E}[p_{jm}] + \sum_{i\prec_m j} x_{im}\mathbb{E}[p_{im}]\Big).$$

We note that given any stochastic (i.e., non-integer) routing matrix $\mathbf{x} \in \mathcal{X}$, it is straightforward to obtain a deterministic routing matrix whose performance is no worse than $V^{\mathrm{R}}(\mathbf{x})$ using the method of conditional probabilities (see, e.g., Motwani and Raghavan 1995).

We will show that with a "smart" choice of the routing matrix, the static routing policy is asymptotically optimal as the number of jobs grows large. We will make this precise in Section 4. We next show how to obtain such a routing matrix by solving a convex quadratic optimization problem.

3.1 Optimal Static Routing

To obtain the optimal static routing matrix, we can, in theory, optimize over the routing matrix directly, i.e., we can solve

$$\min_{\mathbf{x}\in\mathcal{X}} V^{\mathrm{R}}(\mathbf{x}). \tag{1}$$

The expected performance of a static routing policy, $V^{\mathrm{R}}(\mathbf{x})$, is not convex in $\mathbf{x}$ because the quadratic form is not positive semi-definite.[1] As a result, we will instead solve a convex relaxation of (1).

Sethuraman and Squillante (1999) and Skutella (2001) develop convex relaxations for the deterministic version of this problem. This approach involves adding a sufficiently large diagonal

[1]Problem (1) can be formulated as a binary linear program. We show this formulation in Appendix D.

to the quadratic form to make the objective convex and subtracting a commensurate linear term with the net effect of providing a lower bound on the problem. We adapt this method to obtain a convex relaxation to $V^{\mathrm{R}}(\mathbf{x})$. In this result and in later discussions it will be helpful to denote by $\mathbf{R}_m = (r_{ij}^m) \in \mathbb{R}^{J\times J}$ the matrix such that

$$r_{ij}^m = \begin{cases} r_{im} & i = j, \\ r_{im} & j \prec_m i, \\ r_{jm} & i \prec_m j. \end{cases}$$

In addition, we denote the vector of expected processing times on machine m by $\mathbb{E}[\mathbf{p}_m] = (\mathbb{E}[p_{jm}])_{j\in\mathcal{J}} \in \mathbb{R}^J$, and denote by $\mathrm{diag}(\mathbb{E}[\mathbf{p}_m]) \in \mathbb{R}^{J\times J}$ the diagonal matrix whose diagonal entries are $\mathbb{E}[\mathbf{p}_m]$.

Lemma 3.1. *Let $\mathbf{c}_m = \big(w_j\mathbb{E}[p_{jm}]\big)_{j\in\mathcal{J}} \in \mathbb{R}^J$ and $\mathbf{D}_m = \mathrm{diag}(\mathbb{E}[\mathbf{p}_m])\mathbf{R}_m\mathrm{diag}(\mathbb{E}[\mathbf{p}_m]) \in \mathbb{R}^{J\times J}$, and consider the optimization problem*

$$Z^{\mathrm{R}} \triangleq \min_{\mathbf{x}\in\mathcal{X}} \sum_{m\in\mathcal{M}} \frac{1}{2}\mathbf{c}_m'\mathbf{x}_m + \frac{1}{2}\mathbf{x}_m'\mathbf{D}_m\mathbf{x}_m, \tag{2}$$

where $\mathbf{x}_m = \big(x_{jm}\big)_{j\in\mathcal{J}} \in [0,1]^J$. Then (2) is a convex optimization problem; moreover, $Z^{\mathrm{R}} \le V^{\mathrm{R}}(\mathbf{x})$ for all routing matrices $\mathbf{x}\in\mathcal{X}$.

Problem (2) is a convex quadratic optimization problem with $J\times M$ variables and linear constraints that impose the requirement that all jobs must be assigned to one machine. Problem (2) is equivalent to the convex relaxations in Sethuraman and Squillante (1999) and Skutella (2001) when job processing times all equal their expected values. The objective of (2) can be written as $V^{\mathrm{R}}(\mathbf{x}) - \frac{1}{2}\sum_{j\in\mathcal{J}} w_j \sum_{m\in\mathcal{M}} \mathbb{E}[p_{jm}]x_{jm}(1-x_{jm})$, which shows that the lower bound of Lemma 3.1 is tight when the optimal solution of (2) is integral. We let $\mathbf{x}^*$ denote an optimal solution of (2). Using the previous expression, we can relate the expected performance of the associated static routing policy (which we denote simply by V^{R}) to the optimal objective of (2) as

$$V^{\mathrm{R}} \triangleq V^{\mathrm{R}}(\mathbf{x}^*) = Z^{\mathrm{R}} + \frac{1}{2}\sum_{j\in\mathcal{J}} w_j \sum_{m\in\mathcal{M}} x_{jm}^*(1-x_{jm}^*)\mathbb{E}[p_{jm}]\,. \tag{3}$$

In all discussions that follow, unless otherwise stated, when we say "the static routing policy," this should be understood to mean a static routing policy induced by an optimal solution of (2), which will be our main focus. In contrast, an "optimal static routing policy," should be understood to mean a static routing policy induced by an optimal solution of (1).

4 Performance Analysis

In this section, we provide our main results.

Theorem 4.1. *Any static routing policy obtained from an optimal solution of (2) satisfies*

$$V^* \le V^{\mathrm{R}} \le V^* + \frac{1}{2}\sum_{j\in\mathcal{J}} w_j\Big(\frac{M-1}{M}\max_{m\in\mathcal{M}}\mathbb{E}[p_{jm}] + \max_{m\in\mathcal{M}}\frac{\mathbb{V}\mathrm{ar}[p_{jm}]}{\mathbb{E}[p_{jm}]}\Big). \tag{4}$$

In particular, if $\mathbb{V}\mathrm{ar}[p_{jm}]/\mathbb{E}[p_{jm}]^2 \le \Delta$ *holds for all jobs* $j \in \mathcal{J}$ *and machines* $m \in \mathcal{M}$*, then the static routing policy satisfies*

$$V^* \le V^{\mathrm{R}} \le V^* + \Big(\frac{M-1}{2M} + \frac{\Delta}{2}\Big)\sum_{j\in\mathcal{J}} w_j \max_{m\in\mathcal{M}}\mathbb{E}[p_{jm}].$$

We prove Theorem 4.1 by bounding the performance V^{R} of the static routing policy from above in terms of Z^{R} and bounding the optimal performance V^* from below by using a penalized perfect information relaxation. By properly choosing the penalty in the information relaxation, we can connect the lower bound to Z^{R}. The analysis leads to a gap between these bounds that is linear in the number of jobs.

We note that in the case with identical machines, i.e., when p_{jm} does not vary with m for every job $j \in \mathcal{J}$, (4) leads to

$$V^{\mathrm{R}} - V^* \le \frac{1}{2}\sum_{j\in\mathcal{J}} w_j\mathbb{E}[p_j]\left(\frac{M-1}{M} + \frac{\mathbb{V}\mathrm{ar}[p_j]}{\mathbb{E}[p_j]^2}\right).$$

This is nearly identical to the sub-optimality gap established for the WSEPT priority policy in Corollary 4.1 of Möhring et al. (1999), with the difference that the coefficient on the second term is slightly weaker ($\frac{1}{2}$ vs. $\frac{M-1}{2M}$) than the result in Corollary 4.1 of Möhring et al. (1999). In Section 5, we establish a slightly stronger result for a different static routing policy in the case of uniformly related machines; this result reduces to the same gap as in Corollary 4.1 of Möhring et al. (1999)

when machines are identical.

If job processing times are deterministic, then a static routing policy is optimal. Note that in this case the variance term in (4) is zero. Thus we can interpret the first term in the gap in (4) as the performance loss due to employing a convex relaxation to find a good routing. If processing times are stochastic, however, the scheduling problem is more challenging, as scheduling policies may be adaptive to realized processing times. Thus, we can interpret the second term in the gap in (4) as the performance loss due to restricting to static policies, which by definition ignore the evolution of job processing information over time.

If we fix the number of machines M and consider scaling the number of jobs J, then the optimal performance V^* scales quadratically with J, because on average each of the J jobs has to wait for $\Omega(J)$ jobs before being processed. From Theorem 4.1, the gap between the performance of the static routing policy and V^*, however, only scales as $O(J)$, provided job weights, mean processing times, and coefficients of variation of processing times are uniformly bounded as J grows. Theorem 4.1 thus implies that the static routing policy is asymptotically optimal under this scaling. This can in fact be strengthened: asymptotic optimality holds as long as the number of jobs grows faster than the number of machines.

Corollary 4.2. *Suppose* $\underline{w} \leq w_j \leq \bar{w}$, $\underline{p} \leq \mathbb{E}[p_{jm}] \leq \bar{p}$, *and* $\mathbb{V}\mathrm{ar}[p_{jm}]/\mathbb{E}[p_{jm}]^2 \leq \Delta$ *for all jobs* $j \in \mathcal{J}$ *and machines* $m \in \mathcal{M}$, *for some positive constants* $\underline{w}$, $\bar{w}$, $\underline{p}$, $\bar{p}$, *and* Δ. *If* $M/J \xrightarrow{J\to\infty} 0$, *then*

$$\frac{V^{\mathrm{R}} - V^*}{V^*} \xrightarrow{J\to\infty} 0.$$

In the next two subsections, we provide further details on the proof of Theorem 4.1; we now describe some high-level intuition as to why we might expect the static routing policy from problem (2) to perform well when there are many jobs relative to machines. First, as we will show shortly, the performance loss of applying a convex relaxation to find a good static routing relative to the optimal static routing is not substantial when there are many jobs. The question then becomes: why should we expect an optimal static routing policy to perform well at all in this regime? Intuitively, an optimal static policy attempts to optimally balance expected job workload statically across machines when jobs are sequenced at each machine by WSEPT. By committing jobs to machines at the outset, the optimal static policy may suffer relative to an optimal, adaptive policy, which balances workloads dynamically depending on machine availability. When every machine has to process many jobs, however, the expectations used in the static routing policy's objective become good approximations of the realized workload, as the total processing time on each machine will tend to concentrate around its mean. Thus, the benefit of dynamically reacting to the realized workload is small in relative terms and we might expect the optimal static routing policy to perform well, as it optimizes an objective function that provides a very good approximation of the problem.

It is perhaps interesting to note that, although this intuition for the asymptotic optimality result rests on averaging (sums of independent variables concentrating around their mean), we do not invoke any concentration inequalities in establishing the performance bounds.

Remark 4.1 (Infeasible job-machine assignments)**.** The upper bound in (4) involves taking, for each job, the maximum of the expected processing times across machines. In some applications, machines may be specialized and simply unable to process certain jobs - i.e., $\mathbb{E}[p_{jm}] = \infty$ for some (j, m) pairs. In such problems, (4) would be a useless bound. A simple modification, however, corrects for this. Specifically, we let $\mathcal{M}_j \subseteq \mathcal{M}$ denote the set of machines that are feasible to job j, and include the constraints that $x_{jm} = 0$ for all $m \notin \mathcal{M}_j$ for all $j \in \mathcal{J}$ in the definition of the set $\mathcal{X}$ of feasible routing matrices. Since an optimal non-anticipative policy would never use such assignments, these constraints are valid in all bounds we derive. The modified version of (4) is then

$$V^* \leq V^{\mathrm{R}} \leq V^* + \frac{1}{2} \sum_{j \in \mathcal{J}} w_j \Big(\frac{M-1}{M} \max_{m \in \mathcal{M}_j} \mathbb{E}[p_{jm}] + \max_{m \in \mathcal{M}_j} \frac{\mathbb{V}\mathrm{ar}[p_{jm}]}{\mathbb{E}[p_{jm}]} \Big).$$

Asymptotic optimality as in Corollary 4.2 follows under the weaker requirement that the conditions on the processing times' means and coefficients of variation hold for all *feasible* job-machine pairs.

Overview of Proof Theorem 4.1 follows from four key steps:

1. We bound from above the performance V^{R} of the static routing policy in terms of the optimal value Z^{R} of the convex relaxation plus a term that is linear in the number of jobs (Proposition 4.3).
2. We bound from below the performance V^* of an optimal adaptive, non-anticipative policy using a penalized perfect information relaxation. The penalized perfect information relaxation involves relaxing the non-anticipativity constraints (i.e., revealing all job processing

times in advance) and adding to the objective a penalty that punishes violations of the non-anticipativity constraints. The penalty we use has two components:

(a) A *sequencing penalty*, which penalizes the perfect information scheduler for prioritizing, on each machine, jobs that have a high ratio of weight to realized processing time.

(b) A *routing penalty*, which penalizes the perfect information scheduler for routing jobs to machines that have short realized processing time.

We show that with these penalties, the penalized perfect information problem, in every sample path, reduces to a deterministic scheduling problem in which it is optimal to sequence jobs routed to each machine according to WSEPT, plus a term that is linear in the job-machine assignments and depends on the routing penalty (Lemma 4.4).

3. We further bound from below the objective value of the penalized perfect information problems in every sample path in terms of Z^{R}. To do this, we consider a convex relaxation of the resulting deterministic scheduling problem and then dualize the job assignment constraints in this relaxation. Let ν_j^* denote the optimal dual variables for the job assignment constraints in the convex problem (2) used to generate the static routing policy. By using ν_j^* in the routing penalty as well as in the dual variables of this relaxation, we can show that the penalized perfect information problem, in every sample path, is bounded from below by Z^{R} less a term that is linear in the number of jobs (Proposition 4.5). We obtain a lower bound on V^* by taking expectations of this lower bound.

4. By combining the upper bound from step 1 and the lower bound from step 3, we obtain an upper bound on the sub-optimality gap of the static routing policy, i.e., an upper bound on $V^{\mathrm{R}} - V^*$, that is linear in the number of jobs. Under the given conditions on the job weights and processing distributions, this result implies asymptotic optimality of the static routing policy as the number of jobs grows large.

We now move to the details of the proof.

4.1 Upper Bound

We first provide an upper bound for the performance V^{R} of the static routing policy, which of course also bounds the optimal performance V^* from above.

Proposition 4.3. *Any static routing policy obtained by solving (2) satisfies*

$$Z^{\mathrm{R}} \leq V^{\mathrm{R}} \leq Z^{\mathrm{R}} + \frac{M-1}{2M} \sum_{j \in \mathcal{J}} w_j \max_{m \in \mathcal{M}} \mathbb{E}[p_{jm}].$$

Proof. From (3),

$$V^{\mathrm{R}} = Z^{\mathrm{R}} + \frac{1}{2} \sum_{j \in \mathcal{J}} w_j \sum_{m \in \mathcal{M}} x^*_{jm}(1 - x^*_{jm}) \mathbb{E}[p_{jm}] \leq Z^{\mathrm{R}} + \frac{1}{2} \sum_{j \in \mathcal{J}} w_j \max_{m \in \mathcal{M}} \mathbb{E}[p_{jm}] \sum_{m \in \mathcal{M}} x^*_{jm}(1 - x^*_{jm}).$$

Fixing job j, since each x^*_{jm} is non-negative and $\sum_{m \in \mathcal{M}} x^*_{jm} = 1$, we have $\sum_{m \in \mathcal{M}} x^*_{jm}(1 - x^*_{jm}) \leq \frac{M-1}{M}$. As a result,

$$Z^{\mathrm{R}} \leq V^{\mathrm{R}} \leq Z^{\mathrm{R}} + \frac{M-1}{2M} \sum_{j \in \mathcal{J}} w_j \max_{m \in \mathcal{M}} \mathbb{E}[p_{jm}]. \qquad \square$$

Proposition 4.3 states that the gap between Z^{R} and the performance V^{R} of the static routing policy only grows as $O(J)$, provided job weights and expected processing times are uniformly bounded. Since both Z^{R} and V^{R} grow as $\Theta(J^2)$, Proposition 4.3 thus implies Z^{R} will be very close to the actual performance V^{R} when J is large. Moreover, since $Z^{\mathrm{R}} \leq \min_{\mathbf{x} \in \mathcal{X}} V^{\mathrm{R}}(\mathbf{x}) \leq V^{\mathrm{R}}$, when J is large the performance of the static routing policy will be very close to the performance of an optimal static routing policy.

4.2 Lower Bound

In this section we develop a lower bound on V^* by using a penalized perfect information relaxation. Specifically, we consider a decision maker who has access to all realizations of the processing times $\mathbf{p} = \{p_{jm}\}_{j \in \mathcal{J}, m \in \mathcal{M}}$ before making scheduling decisions. Given a sample path $\mathbf{p} \in \mathbb{R}^{J \times M}$ we can calculate the optimal value for the sample path in "hindsight" by solving a deterministic scheduling problem. The expected value with perfect information on processing times provides a lower bound on V^*. Unfortunately, this lower bound is not tight in general because the decision maker with perfect information on processing times can "cheat" (i.e., use information not available to any non-anticipative scheduling policy) by (a) routing jobs to machines that have correspondingly short realized processing times, and then (b) for each machine, by sequencing jobs with higher ratio of weight to realized processing time first. To improve the lower bound we will impose an appropriate penalty that creates incentives for the perfect information scheduler to not "cheat" in the routing and sequencing decisions. In Section 4.3 we show that this penalty is essential to obtain good lower bounds.

Following Brown et al. (2010), we say a penalty Y^π is *dual feasible* if Y^π does not penalize, in expectation, any non-anticipative policy, i.e., if for all $\pi \in \Pi$, $\mathbb{E}[Y^\pi] = 0$ holds.[2] The stochastic scheduling problem including any dual feasible penalty Y^π can be written as

$$V_Y^* = \min_{\pi \in \Pi} \mathbb{E}\Big[\sum_{j \in \mathcal{J}} w_j C_j^\pi + Y^\pi\Big].$$

Since the penalty is dual feasible, $V_Y^* = V^*$. We let $V^{\mathrm{H}}(\mathbf{p})$ denote the optimal (deterministic) value of the penalized perfect information problem for sample path $\mathbf{p} \in \mathbb{R}^{J \times M}$, where H stands for "hindsight." We then have $V^{\mathrm{H}} \triangleq \mathbb{E}[V^{\mathrm{H}}(\mathbf{p})] \leq V_Y^* = V^*$. A dual problem then is to choose a dual feasible penalty so as to maximize the lower bound V^{H}. As shown in Brown et al. (2010), strong duality holds and we can in theory construct an ideal penalty such that $V^{\mathrm{H}} = V^*$ by using the optimal value function to the stochastic scheduling problem. Since the optimal value function is difficult to characterize in general, we will instead consider simpler penalties that will allow us to relate the lower bound V^{H} to the performance of the static routing policy.

Specifically, to compensate for possible cheating at the routing and sequencing steps in the perfect information problem, we consider a penalty of the form $Y^\pi = Y_{\mathrm{S}}^\pi + Y_{\mathrm{R}}^\pi$, where Y_{S}^π is a sequencing penalty and Y_{R}^π is a routing penalty. The goal of the sequencing penalty is to create incentives for the perfect information scheduler to sequence jobs on every machine according to WSEPT in every sample path; the goal of the routing penalty is to create incentives for the perfect information scheduler to route jobs similarly to an optimal solution $\mathbf{x}^*$ of (2) in every sample path. We will also ensure that both penalties are dual feasible.

We set the sequencing penalty Y_{S}^π to be

$$Y_{\mathrm{S}}^\pi \triangleq \sum_{j \in \mathcal{J}} \sum_{m \in \mathcal{M}} r_{jm} S_{jm}^\pi \big(p_{jm} - \mathbb{E}[p_{jm}]\big),$$

[2] Brown et al. (2010) define dual feasible penalties more generally with $\mathbb{E}[Y^\pi] \leq 0$ for all $\pi \in \Pi$, although in most applications of the approach, including here, a penalty has "no slack" in that equality holds in this condition.

where S^{π}_{jm} denotes the starting time of job j on machine m with the convention that $S^{\pi}_{jm} = 0$ if job j is not assigned to machine m. First, note that Y^{π}_{S} is dual feasible: this follows from the fact that, for every non-anticipative policy $\pi \in \Pi$, the processing time p_{jm} of job j on machine m is independent of job j's starting time S^{π}_{jm}. To understand why the sequencing penalty can be helpful, imagine a given job j being routed to machine m in the perfect information problem. If p_{jm} is short (compared to its expected value), to minimize the total weighted completion time, the perfect information scheduler prefers to sequence j earlier; on the other hand, $p_{jm} - \mathbb{E}[p_{jm}] < 0$, so the sequencing penalty creates an incentive to start j later, i.e., to delay the sequencing of j. If p_{jm} is long relative to its expected value, the reverse reasoning applies. With the weights r_{jm} on each term in the sequencing penalty, it turns out that these competing incentives perfectly cancel, and it is optimal in the perfect information problem to sequence all jobs on each machine according to WSEPT in every sample path. We show this formally below.

For the routing penalty Y^{π}_{R}, we consider the form

$$Y^{\pi}_{\mathrm{R}} \triangleq \sum_{j \in \mathcal{J}} \sum_{m \in \mathcal{M}} x^{\pi}_{jm} \Big[\lambda_{jm} \big(\mathbb{E}[p_{jm}] - p_{jm} \big) + \gamma_{jm} \big(\mathbb{E}[p^2_{jm}] - p^2_{jm} \big) \Big],$$

where x^{π}_{jm} indicates whether job $j \in \mathcal{J}$ is assigned to machine $m \in \mathcal{M}$ under policy $\pi \in \Pi$, and $\boldsymbol{\lambda} = (\lambda_{jm})_{j \in \mathcal{J}, m \in \mathcal{M}} \in \mathbb{R}^{J \times M}$ and $\boldsymbol{\gamma} = (\gamma_{jm})_{j \in \mathcal{J}, m \in \mathcal{M}} \in \mathbb{R}^{J \times M}$ are two constant matrices that we will specify shortly. First, Y^{π}_{R} is dual feasible: this follows from the fact that, for every non-anticipative policy $\pi \in \Pi$, the decision about whether to assign job j to machine m is independent of p_{jm}. We can view the routing penalty as creating incentives for the perfect information scheduler to route jobs to machines that have relatively large (compared to their expected values) realized processing times. Note that Y^{π}_{R} is independent of the sequencing of jobs on each machine.

Since both Y^{π}_{S} and Y^{π}_{R} are dual feasible, the full penalty $Y^{\pi} = Y^{\pi}_{S} + Y^{\pi}_{R}$ is also dual feasible. The penalized perfect information problem can be viewed as a combination of an assignment problem together with a deterministic scheduling problem. In this deterministic scheduling problem, weights are proportional to the realized processing times and it is optimal to sequence jobs according to the WSEPT ordering $\prec_m$ on every machine. We now formalize this.

Lemma 4.4. *The penalized perfect information problem with $Y^\pi = Y^\pi_{\mathrm{S}} + Y^\pi_{\mathrm{R}}$ can be written as*

$$V^{\mathrm{H}}(\mathbf{p}) = \min_{\mathbf{x}\in\mathcal{X}} \sum_{j\in\mathcal{J}} \sum_{m\in\mathcal{M}} r_{jm} p_{jm} x_{jm} \Big(p_{jm} + \sum_{i\prec_m j} x_{im} p_{im} \Big)$$
$$+ x_{jm} \Big[(\lambda_{jm} + p_{jm} r_{jm}) \Big(\mathbb{E}[p_{jm}] - p_{jm} \Big) + \gamma_{jm} \Big(\mathbb{E}[p^2_{jm}] - p^2_{jm} \Big) \Big].$$

Proof. We write the completion time as $C_j = \sum_{m\in\mathcal{M}} C_{jm}$ with the assumption that $C_{jm} = 0$ if job j is not assigned to machine m. Using the fact that the starting time is given by $S_{jm} = C_{jm} - p_{jm}x_{jm}$ together with $r_{jm} = w_j/\mathbb{E}[p_{jm}]$, we can write the objective of the penalized perfect information problem as

$$\sum_{j\in\mathcal{J}} w_j C_j + Y_{\mathrm{S}} + Y_{\mathrm{R}} = \sum_{j\in\mathcal{J}} \sum_{m\in\mathcal{M}} r_{jm} p_{jm} C_{jm} + \sum_{j\in\mathcal{J}} \sum_{m\in\mathcal{M}} x_{jm} p_{jm} r_{jm} (\mathbb{E}[p_{jm}] - p_{jm}) + Y_{\mathrm{R}}.$$

The first term corresponds to a *sequencing cost* that coincides with the objective value of the deterministic scheduling problem $R||\sum_{j=1}^{J} \tilde{w}_{jm} C_j$ with weights $\tilde{w}_{jm} = r_{jm}p_{jm}$, while the second and third terms can be interpreted as a *routing cost* that penalizes the perfect information scheduler for assigning jobs to machines in which the realized processing time is below the mean. Note that the routing cost is independent of the order in which jobs are sequenced in a machine, thus once an assignment is fixed the decision maker should sequence jobs so as to minimize the sequencing cost. The ratio of weights to realized processing times in this modified scheduling problem is given by $\tilde{w}_{jm}/p_{jm} = r_{jm}p_{jm}/p_{jm} = r_{jm}$: thus, the optimal ordering is independent of the realized processing times and coincides with the WSEPT rule. As a result, the penalized perfect information problem can be written as in the statement. $\square$

As we do with the static routing policy, we will work with convex relaxations of $V^{\mathrm{H}}(\mathbf{p})$ that are easier to analyze. Following the steps of Lemma 3.1 applied to the deterministic problem $V^{\mathrm{H}}(\mathbf{p})$, we have

$$V^{\mathrm{H}}(\mathbf{p}) \geq Z^{\mathrm{H}}(\mathbf{p}) \triangleq \min_{\mathbf{x}\in\mathcal{X}} \sum_{m\in\mathcal{M}} \mathbf{a}'_m \mathbf{x}_m + \frac{1}{2} \mathbf{x}'_m \mathbf{Q}_m \mathbf{x}_m,$$

where $\mathbf{a}_m = \left(\frac{1}{2} p^2_{jm} r_{jm} + (\lambda_{jm} + p_{jm} r_{jm}) \left(\mathbb{E}[p_{jm}] - p_{jm} \right) + \gamma_{jm} \left(\mathbb{E}[p^2_{jm}] - p^2_{jm} \right) \right)_{j\in\mathcal{J}} \in \mathbb{R}^J$ and $\mathbf{Q}_m = \mathrm{diag}(\mathbf{p}_m) \mathbf{R}_m \mathrm{diag}(\mathbf{p}_m) \in \mathbb{R}^{J\times J}$. Note that the matrix $\mathbf{Q}_m$ is analogous to the matrix $\mathbf{D}_m$ in (2) that we use to compute the static routing policy, with expected job processing times replaced by their realized values. The objective in $Z^{\mathrm{H}}(\mathbf{p})$ is convex, as the matrices $\mathbf{Q}_m$ are positive semi-definite by Lemma 3.1. We denote $Z^{\mathrm{H}} \triangleq \mathbb{E}[Z^{\mathrm{H}}(\mathbf{p})]$. From the above discussion, we have $V^* \geq V^{\mathrm{H}} \geq Z^{\mathrm{H}}$.

We now turn to the issue of how to tune the routing penalty to obtain a good lower bound on V^*. Recall that the goal of the routing penalty is to create incentives for the perfect information scheduler to route jobs in similar fashion to $\mathbf{x}^*$, which is an optimal solution to (2). Since (2) is a convex optimization problem, we can find an optimal Lagrange multiplier ν_j^* on the assignment constraint for each job; ν_j^* represents the marginal cost of assigning a job to any machine at optimality in (2). By choosing λ_{jm} in the routing penalty to be proportional to ν_j^*, we align the marginal cost of routing jobs in the perfect information problem with problem (2), which we use to obtain a static routing policy. As we now show, this allows us to establish a strong relationship between $Z^{\mathrm{H}}(\mathbf{p})$ and Z^{R}, the optimal value of (2), in every sample path.

Proposition 4.5. *Let ν_j^* denote an optimal Lagrange multiplier of the constraint $\sum_{m\in\mathcal{M}} x_{jm} = 1$ in (2). If $\gamma_{jm} = -\frac{1}{2}r_{jm}$ and $\lambda_{jm} = \nu_j^*/\mathbb{E}[p_{jm}] + \frac{1}{2}w_j$, then for every sample path $\mathbf{p}$,*

$$V^{\mathrm{H}}(\mathbf{p}) \geq Z^{\mathrm{H}}(\mathbf{p}) \geq Z^{\mathrm{R}} - \frac{1}{2}\sum_{j\in\mathcal{J}} w_j \max_{m\in\mathcal{M}} \frac{\mathbb{V}\mathrm{ar}[p_{jm}]}{\mathbb{E}[p_{jm}]}. \tag{5}$$

Note that the scaling of ν_j^* by $1/\mathbb{E}[p_{jm}]$ is simply a matching of units: ν_j^* is measured in units of objective (weighted time), whereas λ_{jm} is in units of objective per time. We also note that the second term $\frac{1}{2}w_j$ in λ_{jm} and the choice of γ_{jm} are not critical for the qualitatively important aspect of the result, which is obtaining a gap in (5) that is linear in the number of jobs. These terms are merely for the purposes of cleanly expressing this gap in terms of the coefficients of variation of the job processing times.

Since the penalty is dual feasible, taking expectations over (5), we immediately obtain that

$$V^* \geq V^{\mathrm{H}} \geq Z^{\mathrm{H}} \geq Z^{\mathrm{R}} - \frac{1}{2}\sum_{j\in\mathcal{J}} w_j \max_{m\in\mathcal{M}} \frac{\mathbb{V}\mathrm{ar}[p_{jm}]}{\mathbb{E}[p_{jm}]}.$$

Proposition 4.5 together with Proposition 4.3 thus imply Theorem 4.1 and show that the gap between the optimal performance V^* and the performance of the static routing policy V^{R} only grows as $O(J)$.

In the proof of Proposition 4.5, we bound $Z^{\mathrm{H}}(\mathbf{p})$ from below in terms of Z^{R} by dualizing the assignment constraints in $Z^{\mathrm{H}}(\mathbf{p})$ with Lagrange multipliers ν_j^*. The problem then decouples over machines. Ideally, one would like to show that, for each machine, the optimal static routing vector $\mathbf{x}_m^*$ is optimal for the perfect information problem in every sample path. Instead, we show that by properly perturbing the linear term in the objective of the perfect information problem (which leads to the gap in (5)) and by setting λ_{jm} and γ_{jm} as above, the optimal routing of job j to machine m can be shown to be $x_{jm}^{\mathrm{H}} = x_{jm}^* \mathbb{E}[p_{jm}]/p_{jm}$. This guarantees that the optimal objective value of the static routing policy Z^{R} coincides with the objective value of the dualized and perturbed perfect information problem. This follows because the quadratic forms satisfy $\mathbf{x}_m^{*\prime}\mathbf{D}_m\mathbf{x}_m^* = \mathbf{x}_m^{\mathrm{H}\prime}\mathbf{Q}_m\mathbf{x}_m^{\mathrm{H}}$ (and a similar equality holds for the linear terms). The full proof of Proposition 4.5 and the proof of Corollary 4.2 are given in Appendix A.

4.3 The Importance of Penalties

Both the routing and sequencing penalties are essential in obtaining good lower bounds on the performance of an optimal, non-anticipative scheduling policy: we present two examples illustrating that in the absence of either of these penalties, the perfect information relaxation can be weak (and in particular, does not provide an asymptotically tight bound). The first example excludes the sequencing penalty, and the second example excludes the routing penalty.

Example 4.1 (Role of the sequencing penalty)**.** Consider an example with $M = 1$, and J identical jobs of weight one and processing times p_j having a Bernoulli distribution with mean 0.5. For this single machine scheduling problem, the static routing policy is optimal, and we have

$$V^{\mathrm{R}} = V^* = \sum_{j=1}^{J} \Big(\mathbb{E}[p_j] + \sum_{i<j} \mathbb{E}[p_i] \Big) = \frac{J^2 + J}{4}.$$

Since there is only one machine, the routing penalty has no effect in this example. Without a sequencing penalty, however, a scheduler with perfect information will "cheat" by scheduling first all jobs with zero realized processing times. Let N_1 be the number of jobs with realized processing times of one. The random variable N_1 has binomial distribution with J trials and success probability 0.5. The expected value of the perfect information problem is

$$V^{\mathrm{H}} = \mathbb{E}\Big[\frac{N_1(N_1+1)}{2}\Big] = \frac{J^2 + 3J}{8}\,.$$

For J large, this lower bound is off from V^* by nearly a factor of two.

Example 4.2 (Role of the routing penalty). Consider an example with $M = 2$, and J identical jobs of weight one. The processing times of each job on the first machine are deterministic and equal to $p_{j1} = 0.5$; on the second machine each p_{j2} is given by a Bernoulli distribution with mean 0.5. The static routing policy simply routes each job randomly according to expected processing times, and thus $x_{j1} = x_{j2} = 0.5$ for each job j. This leads to

$$V^{\mathrm{R}} = \sum_{j=1}^{J}\sum_{m=1}^{2}\frac{1}{2}\Big(\mathbb{E}[p_{jm}] + \sum_{i<j}\frac{\mathbb{E}[p_{jm}]}{2}\Big) = \frac{J^2+3J}{8}.$$

Without a routing penalty, a scheduler with perfect information will "cheat" by routing all jobs that have zero realized processing times on the second machine (the random machine) to that machine. This leads to a weak lower bound even with a sequencing penalty. We now show this formally. Without the routing penalty, the objective of $V^{\mathrm{H}}(\mathbf{p})$ (refer to Lemma 4.4) is:

$$\sum_{j\in\mathcal{J}} w_j C_j + Y_{\mathrm{S}} = \sum_{j\in\mathcal{J}}\sum_{m\in\mathcal{M}} r_{jm}p_{jm}C_{jm} + \sum_{j\in\mathcal{J}}\sum_{m\in\mathcal{M}} x_{jm}p_{jm}r_{jm}(\mathbb{E}[p_{jm}] - p_{jm}) \le \sum_{j\in\mathcal{J}}\sum_{m\in\mathcal{M}} r_{jm}p_{jm}C_{jm},$$

where the inequality follows from the fact that $\mathbb{E}[p_{jm}] = p_{jm}$ for the first machine, and for the second machine, either $p_{jm} = 0$ or $p_{jm} = 1$, and in either case $p_{jm}(\mathbb{E}[p_{jm}] - p_{jm}) \le 0$.

Let N_1 be the number of jobs with realized processing times of one on the random machine. The random variable N_1 has a binomial distribution with J trials and success probability 0.5. Clearly, it is optimal to route all other $J - N_1$ jobs to the random machine because they have zero realized job processing times there. The remaining N_1 jobs have "penalized" weights (from the sequencing penalty) $r_{j2}p_{j2} = 2$ and unit realized processing times on the random machine, and unit weights and processing times of 0.5 on the deterministic machine. It is feasible for the perfect information scheduler to route $\frac{1}{5}$ of the N_1 jobs to the random machine and the remaining $\frac{4}{5}$ of the N_1 jobs to the deterministic machine (in fact, this can be shown to be asymptotically optimal when J is large). With this routing in every sample path and the bound above on $\sum_{j\in\mathcal{J}} w_j C_j + Y_{\mathrm{S}}$, we have

$$V^{\mathrm{H}} = \mathbb{E}[V^{\mathrm{H}}(\mathbf{p})] = \mathbb{E}\Big[\min_{\mathbf{x}\in\mathcal{X}}\Big(\sum_{j\in\mathcal{J}} w_j C_j + Y_{\mathrm{S}}\Big)\Big] \le \mathbb{E}\Big[\sum_{j\in\mathcal{J}}\sum_{m\in\mathcal{M}} r_{jm}p_{jm}C_{jm}\Big] = \mathbb{E}\Big[\frac{N_1^2+2N_1}{5}\Big] = \frac{J^2+5J}{20}.$$

For J large, the perfect information bound with a sequencing penalty but no routing penalty in this example is thus off from V^* by at least a factor of $\frac{5}{2}$.

5 Uniformly Related Machines

In this section we consider the case of *uniformly related machines.* Here machines differ only in their speeds: the processing time of job j on machine m is $p_{jm} = p_j/s_m$, where $s_m > 0$ is the speed of machine m. Although this is a special case of the problem considered in Theorem 4.1, using our penalized perfect information relaxation approach, we can establish slightly stronger performance guarantees for a simpler static routing policy. In Section 5.1, we show that similar bounds can also be obtained using LP-based performance space bounds when machines are uniformly related.

Specifically, we consider the static routing policy induced by $x_{jm} = s_m / \sum_{m' \in \mathcal{M}} s_{m'}$, which we refer to as *speed proportional routing.* This routing policy balances job workload across machines by routing jobs proportionally to machine speeds. We note that solving problem (2) when machines are uniformly related does not lead to speed proportional routing, so this policy is different than the static routing policy studied in Section 4. In the following, we let S denote the sum of the speeds and assume that machines are indexed in descending order of their speeds, i.e., $s_1 \geq s_2 \geq \cdots \geq s_M$. We use V^{S} to denote the expected performance of speed proportional routing.

Theorem 5.1. *If machines are uniformly related, then speed proportional routing satisfies*

$$V^* \leq V^{\mathrm{S}} \leq V^* + \frac{1}{2} \sum_{j \in \mathcal{J}} w_j \mathbb{E}[p_j] \Big\{ \Big(\frac{2M-1}{S} - \frac{1}{\kappa_j} \Big) + \Big(\frac{1}{\kappa_j} - \frac{1}{S} \Big) \frac{\mathbb{V}\mathrm{ar}[p_j]}{\mathbb{E}[p_j]^2} \Big\}, \tag{6}$$

where $\kappa_j = s_M$ *if* $\mathbb{V}\mathrm{ar}[p_j]/\mathbb{E}[p_j]^2 > 1$ *and* $\kappa_j = s_1$ *if* $\mathbb{V}\mathrm{ar}[p_j]/\mathbb{E}[p_j]^2 \leq 1$.

If $M = 1$, then from (6) it follows that $V^* = V^{\mathrm{S}}$, which means the performance guarantee in Theorem 5.1 is sharp. If all speeds are equal, then the problem reduces to that of identical parallel machines; without loss of generality we can set $s_m = 1$ in this case and (6) then reduces to

$$V^* \leq V^{\mathrm{S}} \leq V^* + \frac{M-1}{2M} \sum_{j \in \mathcal{J}} w_j \Big(\mathbb{E}[p_j] + \frac{\mathbb{V}\mathrm{ar}[p_j]}{\mathbb{E}[p_j]} \Big).$$

This is the same sub-optimality gap as in Corollary 4.1 in Möhring et al. (1999) for the identical parallel machine problem, albeit as mentioned Möhring et al. (1999) study a different policy. It is also perhaps interesting to note that when $\mathbb{V}\mathrm{ar}[p_j]/\mathbb{E}[p_j]^2 \leq 1$ for all jobs $j \in \mathcal{J}$ (e.g., uniform, exponential, or Erlang distributions), the gap in (6) does not depend on the slowest machine other than through the sum of speeds.

In the proof of Theorem 5.1 we show that speed proportional routing is optimal for a perturbed version of the penalized perfect information problem in which we slightly modify the linear terms of the objective. In the case of uniformly related machines, the routing penalty is not critical to obtain the qualitative result, and the routing penalty is introduced for the purpose of cleanly expressing this gap in terms of the coefficients of variation of the job processing times. Intuitively, because the processing times of a job are perfectly correlated across machines, the perfect information scheduler cannot cheat by routing jobs to machines that have correspondingly short realized processing times. This dependence of processing times across machines allows us to provide a bound for the speed proportional routing policy that is in some cases sharper to the one given in Theorem 4.1. The proof of Theorem 5.1 is given in Appendix A.4.

5.1 LP-Based Performance Space Bounds for Uniformly Related Machines

For stochastic scheduling on identical parallel machines, Möhring et al. (1999) provides a polyhedral relaxation of the performance space, and shows that this polyhedron is a polymatroid; these results build upon similar relaxations developed by Schulz (1996) for deterministic scheduling problems. This relaxation critically relies on job processing times being the same across machines: the intuition is that M identical machines working in parallel tend to process jobs at a similar rate as a single machine that is M times faster. Schulz (1996) derives similar polyhedral relaxations of the performance space for deterministic scheduling on uniformly related machines (see Lemma 9 of Schulz 1996); we are not aware of similar results that extend to the more general case of stochastic scheduling on unrelated machines.

In this subsection, we show that for the case of uniformly related machines, we can in fact derive polyhedral representations of the performance space that lead to similar guarantees (albeit with slightly weaker constants) as in Theorem 5.1. This representation leverages the facts that (i) when machines are uniformly related, the processing times of jobs on different machines will be proportional and (ii) the processing time of any one job can always be bounded from below by the time it would take on the fastest machine. Given this intuition, it is not clear that such results could be extended to the case of unrelated machines.

We have the following result, which parallels Theorem 3.1 in Möhring et al. (1999) in the case of uniformly related machines.

Proposition 5.2. *If machines are uniformly related, then for every feasible, non-anticipative policy, the expected completion times* $\mathbb{E}[C_j]$ *of jobs satisfy, for all* $\mathcal{A} \subseteq \mathcal{J}$*,*

$$\sum_{j\in\mathcal{A}} \mathbb{E}[p_j]\mathbb{E}[C_j] \geq \frac{1}{2S}\Big(\sum_{j\in\mathcal{A}} \mathbb{E}[p_j]\Big)^2 - \frac{1}{2}\Big(\frac{1}{s_M} - \frac{1}{S}\Big)\sum_{j\in\mathcal{A}} \mathbb{V}\mathrm{ar}[p_j] - \Big(\frac{1}{2s_M} - \frac{1}{s_1}\Big)\sum_{j\in\mathcal{A}} \mathbb{E}[p_j]^2\ . \tag{7}$$

As we have assumed above, if processing times satisfy $\mathbb{V}\mathrm{ar}[p_j]/\mathbb{E}[p_j]^2 \leq \Delta$ for all jobs $j \in \mathcal{J}$, then (7) imply that

$$\sum_{j\in\mathcal{A}} \mathbb{E}[p_j]\mathbb{E}[C_j] \geq \frac{1}{2S}\Big(\sum_{j\in\mathcal{A}} \mathbb{E}[p_j]\Big)^2 - \Big(\Big(\frac{1}{2s_M} - \frac{1}{s_1}\Big) + \frac{\Delta}{2}\Big(\frac{1}{s_M} - \frac{1}{S}\Big)\Big)\sum_{j\in\mathcal{A}} \mathbb{E}[p_j]^2\,, \tag{8}$$

holds for all feasible, non-anticipative policies and for all $\mathcal{A} \in \mathcal{J}$. We thus obtain a lower bound on V^* by solving the LP $\min\Big\{\sum_{j\in\mathcal{J}} w_j\mathbb{E}[C_j] \text{ subject to } (8)\Big\}$ treating $\mathbb{E}[C_j]$ as decision variables. It is straightforward to show that the set function on the right-hand side of (8) is supermodular, and thus a solution to this problem is given by the greedy algorithm due to Edmonds (1971). We assume without loss of generality that $\frac{w_1}{\mathbb{E}[p_1]} \geq \frac{w_2}{\mathbb{E}[p_2]} \geq \cdots \geq \frac{w_J}{\mathbb{E}[p_J]}$ and let C^{LP} denote an optimal solution to this problem. Since a greedy solution is optimal, it follows that

$$C_j^{\mathrm{LP}} = \frac{1}{S}\sum_{k=1}^{j} \mathbb{E}[p_k] - \Big(\Big(\frac{1}{2s_M} + \frac{1}{2S} - \frac{1}{s_1}\Big) + \frac{\Delta}{2}\Big(\frac{1}{s_M} - \frac{1}{S}\Big)\Big)\mathbb{E}[p_j].$$

Since $\sum_{j\in\mathcal{J}} w_j C_j^{\mathrm{LP}} \leq V^*$, this leads to a bound on the sub-optimality gap of the speed proportional routing policy:

$$V^{\mathrm{S}} - V^* \leq V^{\mathrm{S}} - \sum_{j\in\mathcal{J}} w_j C_j^{\mathrm{LP}} = \frac{1}{2}\Big(\Big(\frac{1}{s_M} + \frac{2M-1}{S} - \frac{2}{s_1}\Big) + \Big(\frac{1}{s_M} - \frac{1}{S}\Big)\Delta\Big)\sum_{j\in\mathcal{J}} w_j\mathbb{E}[p_j]. \tag{9}$$

The bound in (9) is qualitatively similar to that of Theorem 5.1, and also implies asymptotic optimality of speed proportional routing in the case of uniformly related machines, although the gap in Theorem 5.1 has slightly stronger constants.

One might wonder whether such an analysis with performance space bounds generalizes to the case of unrelated machines. The challenge is that the LP-based bounds on the performance space are, by design, agnostic to the specifics of how policies might assign jobs to machines. Although this may seem like a limitation of this approach, in the case of uniformly related machines, we can nonetheless obtain good (i.e., asymptotically tight) bounds with this approach by leveraging the fact that processing times are proportional across the machines. We do not know, however, how to generalize these LP-based bounds on the performance space in the absence of any particular structure for processing times across machines.

6 Dependent Jobs

The analysis above assumes, consistent with much of the literature on stochastic scheduling, that processing times are independent across jobs. When job processing times are dependent, static policies may perform poorly: good scheduling policies may need to be adaptive, as the elapsed time for jobs in process (or completed) may convey valuable information about the distributions for unscheduled jobs, which could in turn influence scheduling decisions.

In this section, we study the impact that dependence in processing time distributions can have on the sub-optimality of the static routing policy, and provide sufficient conditions for asymptotic optimality of the static routing policy. At a high-level, we use the same proof technique involving penalized perfect information relaxations, but now more care is required in the definition of penalties: for example, with dependent jobs, the start time of a job may no longer be independent of the processing time of the job, and thus the routing and sequencing penalties as defined in Section 4.2 need not be dual feasible. In terms of the results that follow, the conditions that we derive for asymptotic optimality of static routing to prevail in the case of dependent jobs are strong conditions, and we will show by example that these conditions are necessary.

We will describe dependence across jobs in the following way. For a feasible, non-anticipative scheduling policy π, we let $\mathcal{F}_t^\pi$ denote the σ-algebra describing information about processing times at time t, and again S_{jm}^π denotes the start time of job j on machine m under policy π.

Assumption 6.1. *There exist constants α_j and β_j such that*

$$\left|\mathbb{E}[p_{jm}|\mathcal{F}_t^\pi] - \mathbb{E}[p_{jm}]\right| \leq \alpha_j, \tag{10}$$

$$\left|\mathbb{E}[p_{jm}^2|\mathcal{F}_t^\pi] - \mathbb{E}[p_{jm}^2]\right| \leq \beta_j, \tag{11}$$

hold a.s. for every $j \in \mathcal{J}$, $m \in \mathcal{M}$, for all $\pi \in \Pi$, and for all $t \leq S_{jm}^\pi$.

Assumption 6.1 expresses limits on the amount any feasible, non-anticipative scheduling policy can "learn" about each job before it is scheduled from the processing times of jobs that are currently in process or completed. These limits are expressed in terms of maximum absolute deviations α_j and β_j in the first and second moments, respectively, of the job processing time distribution relative to its unconditional (i.e., at $t = 0$) value. Note that the static routing policy (and its performance) only depend on processing time distributions through their expected values, so the dependence structure does not affect V^{R} - but the optimal value V^* may be affected by the dependence structure. As in the case of indepedent jobs, we make no assumptions about the dependence structure of processing times across machines for each job.

In Appendix B.1 we provide a bound (Theorem B.1) on the sub-optimality of the performance of the static routing policy. This result is analogous to Theorem 4.1 in the case of independent jobs, but includes additional terms that depend on α_j and β_j: these terms can be interpreted as bounds on the performance loss of the static routing policy due to ignoring the dependence across jobs. If the dependence across jobs is sufficiently weak, then we might expect the static routing policy to remain asymptotically optimal as the number of jobs grows. The following result makes this precise; this result follow from Theorem B.1 by similar arguments as those in Corollary 4.2.

Corollary 6.2. *Suppose the conditions on weights and processing times in Corollary 4.2 hold. Let $\bar{\alpha} \triangleq \sum_{j\in\mathcal{J}} \alpha_j/J$ and $\bar{\beta} \triangleq \sum_{j\in\mathcal{J}} \beta_j/J$. If $\bar{\alpha}M^{\frac{3}{2}} \xrightarrow{J\to\infty} 0$, $\bar{\beta}M/J \xrightarrow{J\to\infty} 0$, and $M/J \xrightarrow{J\to\infty} 0$, then*

$$\frac{V^{\mathrm{R}} - V^*}{V^*} \xrightarrow{J\to\infty} 0.$$

For a fixed number of machines, the key condition that implies asymptotic optimality of static routing policy is that the average (over jobs) of α_j go to zero as the number of jobs grows large: in other words, it is sufficient that the "average dependence," as measured in terms of absolute deviations of the expected job processing times in (10), vanishes. The condition on the second moments is much weaker; for example, it suffices that the conditional variance of processing times for each job be uniformly bounded for the result to hold. We now show by example that the condition on expected processing times is in fact necessary for the static routing policy to be asymptotically optimal.

Example 6.1 (Jobs with limited dependence). Consider an example with $M = 1$ and J jobs and all job weights are 1. We assume the number of jobs J is even. The processing time p_j of each job j on the machine follows a Bernoulli distribution with mean 0.5. We assume the processing times of every pair of adjacent odd- and even-numbered jobs p_{2i-1} and p_{2i} with $i \in \{1, 2, \cdots, \frac{J}{2}\}$ are dependent, but are independent from all the other jobs. Specifically, the joint distribution of (p_{2i-1}, p_{2i}) is

$\mathbb{P}(p_{2i-1}, p_{2i})$	$p_{2i} = 0$	$p_{2i} = 1$
$p_{2i-1} = 0$	q_i	$\frac{1}{2} - q_i$
$p_{2i-1} = 1$	$\frac{1}{2} - q_i$	q_i

where $q_i \in [0, \frac{1}{2}]$. We have $\mathbb{P}(p_{2i} = 1 | p_{2i-1} = 0) = 1 - 2q_i$ and $\mathbb{P}(p_{2i} = 1 | p_{2i-1} = 1) = 2q_i$. If $q_i = \frac{1}{4}$ for all job pairs (p_{2i-1}, p_{2i}), all jobs are independent and from Section 4 we know the static routing policy is asymptotically optimal as J goes to infinity. If $q_i > \frac{1}{4}$, adjacent jobs are positively correlated. In this example we assume $q_i \geq \frac{1}{4}$ for all job pairs (p_{2i-1}, p_{2i}) and that jobs are sorted in decreasing order of q_i.

Since the (unconditional) expected processing time of each job is the same (0.5), the static routing policy simply chooses an arbitrary order of jobs and processes jobs in this order. Thus,

$$V^{\mathrm{R}} = \sum_{j=1}^{J} \Big(\mathbb{E}[p_j] + \sum_{i<j} \mathbb{E}[p_i] \Big) = \frac{J^2 + J}{4}.$$

Now we consider a simple heuristic policy that is adaptive and exploits the correlation among adjacent jobs. Every time an odd-numbered job $2i - 1$ is completed, if $p_{2i-1} = 0$, then we continue to process job $2i$; otherwise if $p_{2i-1} = 1$, we move job $2i$ to the end of the sequence and process the next odd job $2i + 1$ instead because job $2i$ is more likely to be long as well. When all the odd-numbered jobs are completed, we process the remaining even-numbered jobs in order. Denote by

V^{A} the expected performance of this heuristic adaptive policy. It can be shown (see Appendix B.2) that

$$V^{\mathrm{A}} \le \frac{5}{16}J^2 + \frac{1}{8}J + \left(1 - \frac{J}{2}\right)\sum_{i=1}^{\frac{J}{2}} q_i.$$

Let $\bar{q} = \frac{2}{J}\sum_{i=1}^{\frac{J}{2}} q_i$ be the average of $\{q_i\}_{i=1}^{\frac{J}{2}}$. Since

$$V^{\mathrm{R}} - V^{\mathrm{A}} \ge \frac{\bar{q} - \frac{1}{4}}{4}J^2 - \frac{\bar{q} - \frac{1}{4}}{2}J \tag{12}$$

and $V^{\mathrm{A}} \ge V^*$, the static routing policy is not asymptotically optimal as J increases whenever $\bar{q} > \frac{1}{4}$.

Note that in this example, $\left|\mathbb{E}[p_j|\mathcal{F}_t] - \mathbb{E}[p_j]\right| \le \alpha_j = 2(q_j - \frac{1}{4})$ for all jobs j and times $t \le S_j^\pi$, thus $\bar{\alpha} = 2(\bar{q} - \frac{1}{4})$. In this sense, the assumption on $\bar{\alpha}$ in Corollary 6.2 is necessary (even with a fixed number of machines) for the static routing policy to be asymptotically optimal.

Example 6.1 demonstrates that even a very small amount of dependence across jobs can lead to the static routing policy being asymptotically suboptimal. Even if only a small, fixed fraction of all the job pairs are weakly dependent (e.g., $q_j = \frac{1}{4} + \epsilon$ for some $\epsilon > 0$), and all other jobs are independent, the static routing policy will not be asymptotically optimal. On the other hand, even if not asymptotically optimal, the static routing policy may perform well if the dependence (as measured by the α_j) is small, and Theorem B.1 provides a performance guarantee.

7 Numerical Examples

Although the preceding analysis provides a guarantee on the static routing policy and shows that this policy is asymptotically optimal, it is also helpful to have methods that produce bounds in specific examples. In particular, we have observed in numerical examples that the lower bounds we study through the use of penalized perfect information relaxations often lead to very good lower bounds on the performance of the optimal scheduling policy.

To demonstrate this, we examine the performance of the static routing policy and the penalized perfect information relaxation lower bound on a set of randomly generated examples. We consider instances with number of jobs J increasing linearly from 50 to 1000 with step length 50. We generate job weights as $w_j \sim \mathrm{U}[0.5, 1]$ and expected processing times as $\mathbb{E}[p_{jm}] \sim \mathrm{U}[0.5, 1]$, all i.i.d. We consider 4 cases:

1. $M = 4$, $p_{jm} \sim \mathrm{U}[0, 2\mathbb{E}[p_{jm}]]$,
2. $M = 4$, $p_{jm} \sim \mathrm{Exponential}(1/\mathbb{E}[p_{jm}])$,
3. $M = \sqrt{J}$, $p_{jm} \sim \mathrm{U}[0, 2\mathbb{E}[p_{jm}]]$,
4. $M = \sqrt{J}$, $p_{jm} \sim \mathrm{Exponential}(1/\mathbb{E}[p_{jm}])$.

In cases 1 and 2, M is fixed to be 4; in cases 3 and 4, M scales with J as $M = \sqrt{J}$ (specifically, we round M to the integer closest to $\sqrt{J}$). For consistency across distributions, for each value of J, all four cases share the same w_j, case 1 and case 2 share the same $\mathbb{E}[p_{jm}]$, and case 3 and case 4 share the same $\mathbb{E}[p_{jm}]$.

In each case and for each fixed J, we calculate: (a) the performance of the static routing policy V^{R}; (b) the penalized perfect information relaxation lower bound Z^{H}; and (c) the perfect information relaxation lower bound without penalty Z_0^{H}. As the static routing policy only depends on expected processing times as does the associated performance, V^{R} is the same for each value of J in cases 1 and 2, as well as for each value of J in cases 3 and 4. The values $Z^{\mathrm{H}} = \mathbb{E}[Z^{\mathrm{H}}(\mathbf{p})]$ and $Z_0^{\mathrm{H}} = \mathbb{E}[Z_0^{\mathrm{H}}(\mathbf{p})]$ are estimated with 100 sample paths; this led to very low standard errors in the results. (We also calculated but do not report here bounds with penalized information relaxations that omitted either the sequencing penalty or the routing penalty, respectively. These lower bounds with only one component of the penalty were not asymptotically optimal and were often quite weak - in some cases these bounds were worse than those with no penalty. Following Section 4.3, this further underscores the importance of having both the sequencing and routing penalties.)

All bounds are calculated using Matlab on a desktop PC, with the MOSEK optimization toolbox used to solve the convex quadratic optimization problems. Solving (2) or optimizing $Z^{\mathrm{H}}(\mathbf{p})$ in one sample path are of comparable difficulty. On the smallest examples ($J = 50$, $M = 4$), these optimization problems take about 0.1 seconds to solve; on the largest examples ($J = 1000$, $M = 32$), these optimization problems take about 100 seconds to solve. Considering the complexity of solving for an optimal policy, these run times appear quite modest by comparison and we could likely reduce the total computational effort somewhat in a variety of ways (e.g., variance reduction techniques for the lower bounds, or solution techniques that take advantage of the special assignment structure of the optimization problems).

Figure 1 reports the simulation results for cases 1 and 2, and Figure 2 reports the simulation results for cases 3 and 4. In all these cases, the perfect information bounds with no penalty are quite weak and the relative gaps using these bounds do not appear to converge to zero. The penalized perfect information bounds, however, perform very well and clearly convey the asymptotic optimality of the static routing policy (on Figures 1a and 1b in particular, the penalized perfect information bound is almost indistinguishable from the performance of the policy). The performance gaps are somewhat larger with exponential distributed jobs and when the number of machines scales with the number of jobs. We suspect these larger gaps are largely due to the lower bound being somewhat worse in these cases. With more machines, there are more opportunities for "cheating" in the penalized perfect information problem. Similarly, exponential processing times are considerably more uncertain than uniform processing times, and thus knowing processing times in advance is more valuable in this case.

Overall, the gaps are quite small, and in many of the examples we can conclude that the static routing policy is at worst within a few percent points from optimality. Although we have derived analytical bounds on the suboptimality of static routing policies, it is encouraging to observe that static routing policies perform well on these specific examples.

8 Conclusion

We have studied the problem of stochastic scheduling on unrelated machines with a weighted sum of expected completion times objective. Our main contribution is to bound the loss in using a simple static routing policy relative to an optimal adaptive, non-anticipative policy; the result implies that the static routing policies we consider approach optimality in the asymptotic regime of many jobs. Moreover, the static routing policies we study can be obtained by solving a convex quadratic optimization problem with $J \times M$ variables that only depends on the job processing distributions through their expected values. In the special case of uniformly related machines, we also have established a similar but slightly sharper result for a simple static routing policy; this policy involves no optimization and simply routes jobs proportionally to machine speeds. Finally, we also study the impact that dependence across job processing times can have on the sub-optimality on the static routing policy, and derive some conditions on the dependence under which static routing is still asymptotically optimal; these conditions are relatively strong and, as we show by example, necessary. The main technique underlying our results has been the use of perfect information relaxations with a properly designed penalty.

There are a number of interesting directions to consider moving forward. First, although the problem of stochastic scheduling on unrelated machines that we study is general in many ways, there are of course many important variations that we do not consider, such as release dates, job precedence constraints, and different objectives (e.g., expected makespan). The static routing policies we study clearly require some modifications to handle such variations, and in fact very different policies may be necessary. Nonetheless, we believe the machinery we use for developing lower bounds on optimal performance may be useful for other stochastic scheduling problems with different features. It would also be interesting to see if the information relaxation duality approach would be useful in other scheduling problems, e.g., stochastic job shop models. Second, in some practical applications of the model it may be the case that the problem is too large to solve to optimality, yet too small for the static routing policies we studied here to be nearly optimal. In such situations, it would be interesting to consider the use of more sophisticated policies and information relaxation lower bounds with more sophisticated penalties. Finally, the performance bounds we derived can be a useful building block in analyzing broader "design" type questions, such as the problem of selecting the best set of machines to process a fixed set of jobs to minimize a combined objective of machine cost plus expected completion time. The implications our results have for variations on this theme (e.g., scheduling problems that include job selection as a decision) may also be interesting to consider.

Acknowledgements

We thank our colleague Jim Smith, the area editor, an anonymous associate editor, and two anonymous referees for providing a number of helpful comments that greatly improved the paper.

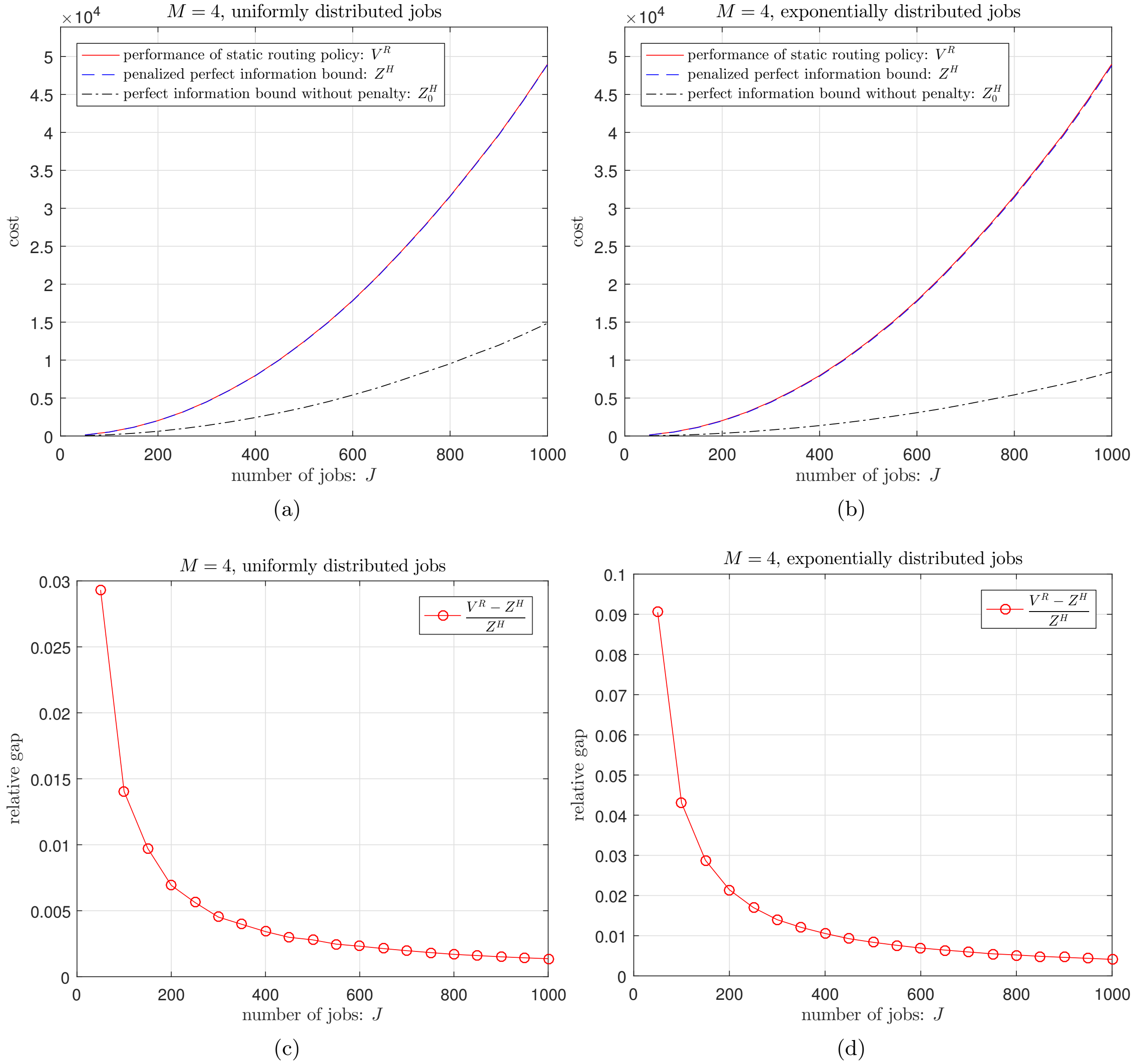

Figure 1: Simulation results with $M = 4$. Jobs are uniformly distributed in (a) and (c), and are exponentially distributed in (b) and (d). Standard errors of Z^{H} and Z_0^{H} are not included in the plots but they are negligible relative to the cost. The penalized perfect information bound is almost indistinguishable from the performance of the policy in (a) and (b).

References

Agrawal, A., Klein, P. and Ravi, R. (1995), 'When trees collide: An approximation algorithm for the generalized steiner problem on networks', *SIAM Journal on Computing* **24**(3), 440–456.

Anand, S., Garg, N. and Kumar, A. (2012), Resource augmentation for weighted flow-time explained by dual fitting, *in* 'Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms', pp. 1228–1241.

Balseiro, S. R. and Brown, D. B. (2016), 'Approximations to stochastic dynamic programs via

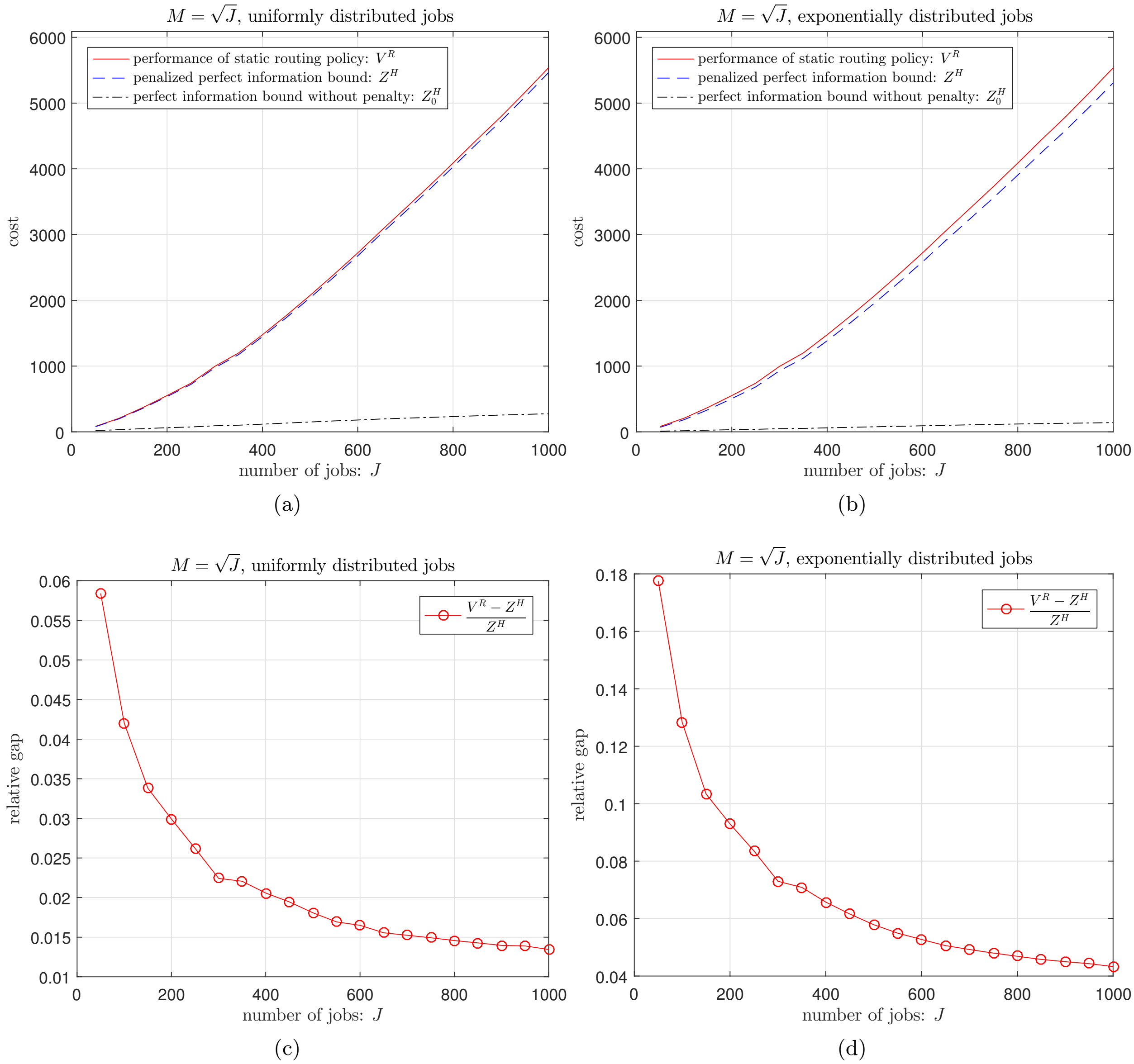

Figure 2: Simulation results with $M = \sqrt{J}$. Jobs are uniformly distributed in (a) and (c), and are exponentially distributed in (b) and (d). Standard errors of Z^{H} and Z_0^{H} are not included in the plots but they are negligible relative to the cost.

information relaxation duality'. Working paper.

Bar-Yehuda, R. and Even, S. (1981), 'A linear-time approximation algorithm for the weighted vertex cover problem', *Journal of Algorithms* **2**(2), 198–203.

Bertsekas, D. P. (1999), *Nonlinear Programming*, Athena Scientific, Belmont, MA.

Bertsimas, D., Gamarnik, D. and Sethuraman, J. (2003), 'From fluid relaxations to practical algorithms for high-multiplicity job-shop scheduling: The holding cost objective', *Operations Research* **51**(5), 798–813.

Brown, D. B., Smith, J. E. and Sun, P. (2010), 'Information relaxations and duality in stochastic dynamic programs', *Operations Research* **58**(4-part-1), 785–801.

Dai, J. G. (1995), 'On positive harris recurrence of multiclass queueing networks: A unified approach via fluid limit models', *The Annals of Applied Probability* **5**(1), 49–77.

Dai, J. G. and Lin, W. (2005), 'Maximum pressure policies in stochastic processing networks', *Operations Research* **53**(2), 197–218.

Dai, J. and Lin, W. (2008), 'Asymptotic optimality of maximum pressure policies in stochastic processing networks', *The Annals of Applied Probability* **18**(6), 2239–2299.

Edmonds, J. (1971), 'Matroids and the greedy algorithm', *Mathematical Programming* **1**(1), 127–136.

Graham, R. L., Lawler, E. L., Lenstra, J. K. and Rinnooy Kan, A. H. G. (1979), 'Optimization and approximation in deterministic sequencing and scheduling: a survey', *Annals of discrete mathematics* **5**(2), 287–326.

Gupta, V., Moseley, B., Uetz, M. and Xie, Q. (2017), 'Stochastic online scheduling on unrelated machines', *Integer Programming and Combinatorial Optimization (IPCO)* .

Hall, L. A., Schulz, A. S., Shmoys, D. B. and Wein, J. (1997), 'Scheduling to minimize average completion time: Off-line and on-line approximation algorithms', *Mathematics of Operations Research* **22**(3), 513–544.

Haugh, M. B. and Kogan, L. (2004), 'Pricing American options: A duality approach', *Operations Research* **52**, 258–270.

Hoogeveen, H., Schuurman, P. and Woeginger, G. (2001), 'Non-approximability results for scheduling problems with minsum criteria', *INFORMS J. on Computing* **13**(2), 157–168.

Jain, K. and Vazirani, V. V. (2001), 'Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation', *Journal of the ACM (JACM)* **48**(2), 274–296.

Maglaras, C. (2000), 'Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality', *Annals of Applied Probability* pp. 897–929.

Möhring, R. H., Schulz, A. S. and Uetz, M. (1999), 'Approximation in stochastic scheduling: The power of lp-based priority policies', *J. ACM* **46**(6), 924–942.

Motwani, R. and Raghavan, P. (1995), *Randomized Algorithms*, Cambridge University Press, New York, NY, USA.

Nazarathy, Y. and Weiss, G. (2009), 'Near optimal control of queueing networks over a finite time horizon', *Annals of Operations Research* **170**(1), 233–249.

Rogers, L. (2002), 'Monte carlo valuation of American options', *Mathematical Finance* **12**, 271–286.

Rothkopf, M. (1966), 'Scheduling with random service times', *Management Science* **12**, 703–713.

Scharbrodt, M., Schickinger, T. and Steger, A. (2006), 'A new average case analysis for completion time scheduling', *J. ACM* **53**(1), 121–146.

Schulz, A. S. (1996), *Scheduling to minimize total weighted completion time: Performance guarantees of LP-based heuristics and lower bounds*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 301–315.

Schulz, A. S. and Skutella, M. (2002), 'Scheduling unrelated machines by randomized rounding', *SIAM Journal on Discrete Mathematics* **15**(4), 450–469.

Schulz, A. S. et al. (1996), Polytopes and scheduling, PhD thesis, Technische Universität Berlin.

Sethuraman, J. and Squillante, M. S. (1999), 'Optimal stochastic scheduling in multiclass parallel queues', *ACM SIGMETRICS Performance Evaluation Review* **27**(1), 93–102.

Skutella, M. (2001), 'Convex quadratic and semidefinite programming relaxations in scheduling', *Journal of the ACM* **48**(2), 206–242.

Skutella, M., Sviridenko, M. and Uetz, M. (2016), 'Unrelated machine scheduling with stochastic processing times', *Mathematics of Operations Research* **41**(3), 851–864.

Smith, W. (1956), 'Various optimizers for single-stage production', *Nav. Res. Log. Quarterly* **3**, 59–66.

Souza, A. and Steger, A. (2006), 'The expected competitive ratio for weighted completion time scheduling', *Theory of Computing Systems* **39**(1), 121–136.

Uetz, M. (2003), 'When greediness fails: Examples from stochastic scheduling', *Oper. Res. Lett.* **31**(6), 413–419.

A Proofs

A.1 Proof of Lemma 3.1

This result is established in both Sethuraman and Squillante (1999) and Skutella (2001); we provide a proof here for completeness. Let $\tilde{\mathbf{D}}_m = (\tilde{d}_{ij}^m) \in \mathbb{R}^{J\times J}$ be the matrix such that

$$\tilde{d}_{ij}^m = \begin{cases} 0\,, & \text{if } i = j\,, \\ w_i \mathbb{E}[p_{jm}]\,, & \text{if } j \prec_m i\,, \\ w_j \mathbb{E}[p_{im}]\,, & \text{if } i \prec_m j\,. \end{cases}$$

The objective in (1) can be written as

$$\begin{aligned} \sum_{m\in\mathcal{M}} \mathbf{c}_m'\mathbf{x}_m + \frac{1}{2}\mathbf{x}_m'\tilde{\mathbf{D}}_m\mathbf{x}_m &= \sum_{m\in\mathcal{M}} \mathbf{c}_m'\mathbf{x}_m - \frac{1}{2}\mathbf{x}_m'\mathrm{diag}(\mathbf{c}_m)\mathbf{x}_m + \frac{1}{2}\mathbf{x}_m'\left(\tilde{\mathbf{D}}_m + \mathrm{diag}(\mathbf{c}_m)\right)\mathbf{x}_m \\ &\geq \sum_{m\in\mathcal{M}} \frac{1}{2}\mathbf{c}_m'\mathbf{x}_m + \frac{1}{2}\mathbf{x}_m'\left(\tilde{\mathbf{D}}_m + \mathrm{diag}(\mathbf{c}_m)\right)\mathbf{x}_m\,, \end{aligned}$$

where we use the fact that $x_{jm}^2 \leq x_{jm}$ because the variables take values in the interval $[0,1]$. It is not hard to check that the matrix $\mathbf{D}_m = \tilde{\mathbf{D}}_m + \mathrm{diag}(\mathbf{c}_m)$ is positive semi-definite, which implies that problem (2) is a convex optimization problem.

A.2 Proof of Proposition 4.5

Define the Lagrangian function of problem (2) to be

$$L(\mathbf{x},\boldsymbol{\nu}) = \boldsymbol{\nu}'\mathbf{1} + \sum_{m=1}^{M} \frac{1}{2}\mathbf{x}_m'\mathbf{D}_m\mathbf{x}_m + \left(\frac{\mathbf{c}_m}{2} - \boldsymbol{\nu}\right)'\mathbf{x}_m,$$

where $\boldsymbol{\nu} \in \mathbb{R}^J$, $\mathbf{x}_m \in \mathbb{R}^J$, and $\mathbf{x} = (\mathbf{x}_m)_{m\in\mathcal{M}} \in \mathbb{R}^{J\times M}$. Since problem (2) is convex and all its constraints are linear, strong duality holds. Moreover, let $\mathbf{x}^* \in \mathbb{R}^{J\times M}$ be an optimal solution to problem (2) and $\boldsymbol{\nu}^* \in \mathbb{R}^J$ be an optimal Lagrange multiplier of the assignment constraint $\sum_{m\in\mathcal{M}} \mathbf{x}_m = \mathbf{1}$ in (2). Because the optimal objective value is finite, Proposition 5.2.1 in Bertsekas (1999) implies that the optimal solution $\mathbf{x}^*$ minimizes the Lagrangian evaluated at $\boldsymbol{\nu}^*$, or equivalently

$$Z^{\mathrm{R}} = L(\mathbf{x}^*, \boldsymbol{\nu}^*) = \min_{\mathbf{x}\geq\mathbf{0}} L(\mathbf{x}, \boldsymbol{\nu}^*). \tag{13}$$

Recall that

$$Z^{\mathrm{H}}(\mathbf{p}) = \min_{\mathbf{x}\in\mathcal{X}} \sum_{m\in\mathcal{M}} \mathbf{a}_m'\mathbf{x}_m + \frac{1}{2}\mathbf{x}_m'\mathbf{Q}_m\mathbf{x}_m,$$

with vectors $\mathbf{a}_m = \left(\frac{1}{2}p_{jm}^2 r_{jm} + (\lambda_{jm} + p_{jm}r_{jm})\left(\mathbb{E}[p_{jm}] - p_{jm}\right) + \gamma_{jm}\left(\mathbb{E}[p_{jm}^2] - p_{jm}^2\right)\right)_{j\in\mathcal{J}} \in \mathbb{R}^J$ and matrices $\mathbf{Q}_m = \mathrm{diag}(\mathbf{p}_m)\mathbf{R}_m\mathrm{diag}(\mathbf{p}_m) \in \mathbb{R}^{J\times J}$. For ease of analysis in the future, we instead consider the following problem:

$$\tilde{Z}^{\mathrm{H}}(\mathbf{p}) = \min_{\mathbf{x}\in\mathcal{X}} \sum_{m\in\mathcal{M}} \tilde{\mathbf{a}}_m'\mathbf{x}_m + \frac{1}{2}\mathbf{x}_m'\mathbf{Q}_m\mathbf{x}_m,$$

with vectors $\tilde{\mathbf{a}}_m = \left((\lambda_{jm} - \frac{1}{2}w_j)\left(\mathbb{E}[p_{jm}] - p_{jm}\right) + \frac{1}{2}w_j p_{jm}\right)_{j\in\mathcal{J}}$. Setting $\gamma_{jm} = -\frac{1}{2}r_{jm}$, a direct relation between $Z^{\mathrm{H}}(\mathbf{p})$ and $\tilde{Z}^{\mathrm{H}}(\mathbf{p})$ can be given by

$$\tilde{Z}^{\mathrm{H}}(\mathbf{p}) - Z^{\mathrm{H}}(\mathbf{p}) \leq \sum_{j\in\mathcal{J}} \max_{m\in\mathcal{M}} (\tilde{a}_{jm} - a_{jm}) = \frac{1}{2}\sum_{j\in\mathcal{J}} \max_{m\in\mathcal{M}} r_{jm}\mathbb{V}\mathrm{ar}[p_{jm}]. \tag{14}$$

We now bound $\tilde{Z}^{\mathrm{H}}(\mathbf{p})$ from below. Let $\mathbf{w} = (w_j)_{j\in\mathcal{J}} \in \mathbb{R}^J$. We set $\boldsymbol{\lambda}_m$ to be

$$\boldsymbol{\lambda}_m = \mathrm{diag}(\mathbb{E}[\mathbf{p}_m])^{-1}\boldsymbol{\nu}^* + \frac{1}{2}\mathbf{w}.$$

As a result we have

$$\tilde{\mathbf{a}}_m - \boldsymbol{\nu}^* = -\mathrm{diag}(\mathbf{p}_m)(\boldsymbol{\lambda}_m - \mathbf{w}). \tag{15}$$

Define new variables $\tilde{\mathbf{x}}_m$ such that if $p_{jm} > 0$ then $p_{jm}x_{jm} = \mathbb{E}[p_{jm}]\tilde{x}_{jm}$; otherwise, $\tilde{x}_{jm}$ can be an arbitrary number. Therefore, for every sample path

$$
\begin{aligned}
\tilde{Z}^{\mathrm{H}}(\mathbf{p}) &\overset{\text{(i)}}{\geq} \boldsymbol{\nu}^{*\prime}\mathbf{1} + \sum_{m=1}^{M} \min_{\mathbf{x}_m \geq \mathbf{0}} \frac{1}{2}\mathbf{x}_m'\mathbf{Q}_m\mathbf{x}_m + (\tilde{\mathbf{a}}_m - \boldsymbol{\nu}^*)'\mathbf{x}_m \\
&\overset{\text{(ii)}}{=} \boldsymbol{\nu}^{*\prime}\mathbf{1} + \sum_{m=1}^{M} \min_{\mathbf{x}_m \geq \mathbf{0}} \frac{1}{2}\mathbf{x}_m'\mathbf{Q}_m\mathbf{x}_m - (\boldsymbol{\lambda}_m - \mathbf{w})'\mathrm{diag}(\mathbf{p}_m)\mathbf{x}_m \\
&= \boldsymbol{\nu}^{*\prime}\mathbf{1} + \sum_{m=1}^{M} \min_{\mathbf{x}_m \geq \mathbf{0}} \frac{1}{2}\mathbf{x}_m'\mathrm{diag}(\mathbf{p}_m)\mathbf{R}_m\mathrm{diag}(\mathbf{p}_m)\mathbf{x}_m - (\boldsymbol{\lambda}_m - \mathbf{w})'\mathrm{diag}(\mathbf{p}_m)\mathbf{x}_m \\
&\overset{\text{(iii)}}{\geq} \boldsymbol{\nu}^{*\prime}\mathbf{1} + \sum_{m=1}^{M} \min_{\tilde{\mathbf{x}}_m \geq \mathbf{0}} \frac{1}{2}\tilde{\mathbf{x}}_m'\mathrm{diag}(\mathbb{E}[\mathbf{p}_m])\mathbf{R}_m\mathrm{diag}(\mathbb{E}[\mathbf{p}_m])\tilde{\mathbf{x}}_m - (\boldsymbol{\lambda}_m - \mathbf{w})'\mathrm{diag}(\mathbb{E}[\mathbf{p}_m])\tilde{\mathbf{x}}_m \\
&= \boldsymbol{\nu}^{*\prime}\mathbf{1} + \sum_{m=1}^{M} \min_{\tilde{\mathbf{x}}_m \geq \mathbf{0}} \frac{1}{2}\tilde{\mathbf{x}}_m'\mathbf{D}_m\tilde{\mathbf{x}}_m + \left(\frac{\mathbf{c}_m}{2} - \boldsymbol{\nu}^*\right)'\tilde{\mathbf{x}}_m \\
&\overset{\text{(iv)}}{=} Z^{\mathrm{R}},
\end{aligned} \tag{16}
$$

where (i) follows from weak duality, (ii) from (15), (iii) from the change of variables, and (iv) from (13) because the optimal solution $\mathbf{x}^*$ minimizes the Lagrangian evaluated at $\boldsymbol{\nu}^*$. Combining (14) and (16), we have for every sample path

$$
V^{\mathrm{H}}(\mathbf{p}) \geq Z^{\mathrm{H}}(\mathbf{p}) \geq Z^{\mathrm{R}} - \frac{1}{2}\sum_{j\in\mathcal{J}} w_j \max_{m\in\mathcal{M}} \frac{\mathbb{V}\mathrm{ar}[p_{jm}]}{\mathbb{E}[p_{jm}]}.
$$

A.3 Proof of Corollary 4.2

Clearly the ratio is non-negative. Here we bound the ratio from above and then show that the upper bound converges to zero as the total number of jobs J goes to infinity. From Theorem 4.1, the difference between V^{R} and V^* can be bounded from above by

$$
V^{\mathrm{R}} - V^* \leq \frac{M(\Delta+1)-1}{2M}\bar{w}\bar{p}J. \tag{17}
$$

On the other hand, according to Proposition 4.5, the optimal performance V^* can be bounded from below by

$$
V^* \geq Z^{\mathrm{R}} - \frac{\Delta}{2}\sum_{j\in\mathcal{J}} w_j \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}] \geq Z^{\mathrm{R}} - \frac{\Delta}{2}\bar{w}\bar{p}J. \tag{18}
$$

Comparing with (2), Z^{R} can further be bounded from below by $\underline{Z}^{\mathrm{R}}$, which is the optimal value of the following convex optimization problem

$$
\underline{Z}^{\mathrm{R}} \triangleq \min_{\mathbf{x}\in\mathcal{X}} \sum_{m\in\mathcal{M}} \frac{1}{2}\underline{\mathbf{c}}'\mathbf{x}_m + \frac{1}{2}\mathbf{x}_m'\underline{\mathbf{D}}\mathbf{x}_m, \tag{19}
$$

with $\underline{\mathbf{c}} = \left(\underline{wp}\right)_{j\in\mathcal{J}} \in \mathbb{R}^J$, and $\underline{\mathbf{D}} = (\underline{d}_{ij}) \in \mathbb{R}^{J\times J}$ the matrix such that $\underline{d}_{ij} = \underline{wp}$. The optimal value of (19) provides a lower bound on Z^{R} because $x_{jm} \geq 0$, $\underline{c}_j \leq c_{jm}$, and $\underline{d}_{ij} \leq d_{ij}^m$. Because this new problem is symmetric across machines, the optimal solution of (19) is $x_{jm} = \frac{1}{M}$ (this can be verified easily by checking the KKT conditions of (19)). As a result,

$$Z^{\mathrm{R}} \geq \underline{Z}^{\mathrm{R}} = \frac{1}{2}\underline{wp}\left(\frac{J^2}{M} + J\right). \tag{20}$$

Combining (18) and (20) we obtain that

$$V^* \geq \frac{1}{2}\underline{wp}\frac{J^2}{M} - \frac{\Delta}{2}\bar{w}\bar{p}J. \tag{21}$$

The right hand side of (21) is positive when J is large enough. Finally, combining (17) and (21) together with the fact that $M/J \xrightarrow{J\to\infty} 0$ yields $\frac{V^{\mathrm{R}}-V^*}{V^*} \xrightarrow{J\to\infty} 0$.

A.4 Proof of Theorem 5.1

Here we use the same form for the sequencing and routing penalties as in the proof of Theorem 4.1, but we use different values for the parameters $(\lambda_{jm})_{j\in\mathcal{J},m\in\mathcal{M}}$ in the routing penalty.

Let $\mathbf{c} = \big(w_j\mathbb{E}[p_j]\big)_{j\in\mathcal{J}} \in \mathbb{R}^J$ and $\mathbf{D} = (d_{ij}) \in \mathbb{R}^{J\times J}$ be the matrix such that

$$d_{ij} = \begin{cases} w_i\mathbb{E}[p_i]\,, & \text{if } i = j\,, \\ w_i\mathbb{E}[p_j]\,, & \text{if } j \prec i\,, \\ w_j\mathbb{E}[p_i]\,, & \text{if } i \prec j\,. \end{cases}$$

Then $\mathbf{c}_m = \frac{1}{s_m}\mathbf{c}$ and $\mathbf{D}_m = \frac{1}{s_m}\mathbf{D}$ because machines are uniformly related. Set the objective function $\bar{Z}^{\mathrm{R}}(\mathbf{x})$ to be

$$\bar{Z}^{\mathrm{R}}(\mathbf{x}) = \sum_{m\in\mathcal{M}} \frac{1}{2}\mathbf{c}'\mathbf{x}_m + \frac{1}{2}\mathbf{x}_m'\mathbf{D}_m\mathbf{x}_m.$$

It is easy to check that the performance of the static routing policy with routing $\mathbf{x} \in \mathcal{X}$ satisfies

$$V^{\mathrm{R}}(\mathbf{x}) = \bar{Z}^{\mathrm{R}}(\mathbf{x}) + \sum_{m\in\mathcal{M}} \frac{1}{2}(\mathbf{c}_m - \mathbf{c})'\mathbf{x}_m + \frac{1}{2}\sum_{j\in\mathcal{J}} w_j \sum_{m\in\mathcal{M}} x_{jm}(1-x_{jm})\mathbb{E}[p_{jm}]. \tag{22}$$

Now consider the optimization problem:

$$\bar{Z}^{\mathrm{R}} = \min_{\mathbf{x}\in\mathcal{X}} \bar{Z}^{\mathrm{R}}(\mathbf{x}). \tag{23}$$

Since problem (23) is convex and all its constraints are linear, according to the KKT conditions, there exists vectors $\mathbf{x}_m^* \in \mathbb{R}^J$, $\boldsymbol{\nu}^* \in \mathbb{R}^J$, and $\boldsymbol{\mu}_m^* \in \mathbb{R}^J$ such that

$$\begin{aligned} \sum_{m\in\mathcal{M}} \mathbf{x}_m^* &= \mathbf{1}, \\ \mathbf{x}_m^* &\geq 0, \\ \boldsymbol{\mu}_m^* &\geq 0, \\ \mathbf{D}_m\mathbf{x}_m^* + \frac{\mathbf{c}}{2} &= \boldsymbol{\nu}^* + \boldsymbol{\mu}_m^*, \\ {\mathbf{x}_m^*}'\boldsymbol{\mu}_m^* &= 0, \end{aligned}$$

with $\boldsymbol{\nu}^*$ and $\boldsymbol{\mu}_m^*$ being the duals corresponding to the normalization and non-negativity constraints of (23), respectively. Furthermore, $\mathbf{x}_m^*$ is an optimal solution of (23), and $\boldsymbol{\nu}^*$ and $\boldsymbol{\mu}_m^*$ is an optimal solution of the dual problem of (23). By checking the KKT conditions, it is easy to verify that the speed proportional routing $x_{jm}^* = s_m/S$ is an optimal solution of (23). The performance V^{S} of speed proportional routing satisfies $V^{\mathrm{S}} = V^{\mathrm{R}}(\mathbf{x}^*)$. According to (22),

$$V^{\mathrm{S}} = \bar{Z}^{\mathrm{R}} + \frac{2M - S - 1}{2S} \sum_{j \in \mathcal{J}} w_j \mathbb{E}[p_j]. \tag{24}$$

Now set $Z^{\mathrm{H}} = \mathbb{E}[Z^{\mathrm{H}}(\mathbf{p})]$, where

$$Z^{\mathrm{H}}(\mathbf{p}) = \min_{\mathbf{x} \in \mathcal{X}} \sum_{m \in \mathcal{M}} \mathbf{a}_m' \mathbf{x}_m + \frac{1}{2} \mathbf{x}_m' \mathbf{Q}_m \mathbf{x}_m,$$

with vectors $\mathbf{a}_m = \left(\frac{1}{2} p_{jm}^2 r_{jm} + (\lambda_{jm} + p_{jm} r_{jm}) \left(\mathbb{E}[p_{jm}] - p_{jm}\right) + \gamma_{jm}\left(\mathbb{E}[p_{jm}^2] - p_{jm}^2\right)\right)_{j \in \mathcal{J}} \in \mathbb{R}^J$ and matrices $\mathbf{Q}_m = \operatorname{diag}(\mathbf{p}_m) \mathbf{R}_m \operatorname{diag}(\mathbf{p}_m) \in \mathbb{R}^{J \times J}$. As in SubSection 4.2, Z^{H} is a lower bound on V^*. Consider the following problem:

$$\tilde{Z}^{\mathrm{H}}(\mathbf{p}) = \min_{\mathbf{x} \in \mathcal{X}} \sum_{m \in \mathcal{M}} \tilde{\mathbf{a}}_m' \mathbf{x}_m + \frac{1}{2} \mathbf{x}_m' \mathbf{Q}_m \mathbf{x}_m,$$

with vectors $\tilde{\mathbf{a}}_m = \left((\lambda_{jm} - w_j)\left(\mathbb{E}[p_{jm}] - p_{jm}\right) + \frac{1}{2} c_j\right)_{j \in \mathcal{J}}$. Setting $\gamma_{jm} = -\frac{1}{2} r_{jm}$, we have

$$\tilde{a}_{jm} - a_{jm} = \frac{1}{2}(c_j - c_{jm}) + \frac{1}{2} r_{jm} \mathbb{V}\mathrm{ar}[p_{jm}] = \frac{1}{2} w_j \mathbb{E}[p_j] \left[1 + \frac{1}{s_m} \left(\frac{\mathbb{V}\mathrm{ar}[p_j]}{\mathbb{E}[p_j]^2} - 1 \right)\right].$$

Therefore, a direct relation between $Z^{\mathrm{H}}(\mathbf{p})$ and $\tilde{Z}^{\mathrm{H}}(\mathbf{p})$ is given by

$$\tilde{Z}^{\mathrm{H}}(\mathbf{p}) - Z^{\mathrm{H}}(\mathbf{p}) \le \sum_{j \in \mathcal{J}} \max_{m \in \mathcal{M}} (\tilde{a}_{jm} - a_{jm}) = \frac{1}{2} \sum_{j \in \mathcal{J}} w_j \mathbb{E}[p_j] \max_{m \in \mathcal{M}} \left[1 + \frac{1}{s_m} \left(\frac{\mathbb{V}\mathrm{ar}[p_j]}{\mathbb{E}[p_j]^2} - 1 \right)\right].$$

Hence, letting $\tilde{Z}^{\mathrm{H}} = \mathbb{E}[\tilde{Z}^{\mathrm{H}}(\mathbf{p})]$, we have

$$\tilde{Z}^{\mathrm{H}} - Z^{\mathrm{H}} \le \frac{1}{2} \sum_{j \in \mathcal{J}} w_j \mathbb{E}[p_j] \max_{m \in \mathcal{M}} \left[1 + \frac{1}{s_m} \left(\frac{\mathbb{V}\mathrm{ar}[p_j]}{\mathbb{E}[p_j]^2} - 1 \right)\right]. \tag{25}$$

Set the penalty $\boldsymbol{\lambda}_m$ to be $\lambda_{jm} = w_j$. Similarly, it is easy to verify by using the KKT conditions that the speed proportional routing $x_{jm}^* = s_m/S$ is an optimal solution of $\tilde{Z}^{\mathrm{H}}(\mathbf{p})$ for every sample path $\mathbf{p}$. As a result,

$$\begin{aligned}\mathbb{E}[\tilde{Z}^{\mathrm{H}}(\mathbf{p})] &= \mathbb{E}\left[\sum_{m\in\mathcal{M}} \tilde{\mathbf{a}}_m' \mathbf{x}_m^* + \frac{1}{2}\mathbf{x}_m^{*\prime}\mathbf{Q}_m\mathbf{x}_m^*\right] \\ &= \sum_{m\in\mathcal{M}} \frac{1}{2}\mathbf{c}'\mathbf{x}_m^* + \frac{1}{2}\mathbf{x}_m^{*\prime}\mathbf{D}_m\mathbf{x}_m^* + \sum_{m\in\mathcal{M}} \frac{1}{2}\mathbf{x}_m^{*\prime}\big(\mathbb{E}[\mathbf{Q}_m] - \mathbf{D}_m\big)\mathbf{x}_m^* \\ &= \bar{Z}^{\mathrm{R}} + \frac{1}{2}\sum_{m\in\mathcal{M}}\sum_{j\in\mathcal{J}} \frac{w_j\mathbb{E}[p_j]}{s_m}\frac{\mathbb{V}\mathrm{ar}[p_j]}{\mathbb{E}[p_j]^2}\big(x_{jm}^*\big)^2 \\ &= \bar{Z}^{\mathrm{R}} + \frac{1}{2S}\sum_{j\in\mathcal{J}} w_j\mathbb{E}[p_j]\frac{\mathbb{V}\mathrm{ar}[p_j]}{\mathbb{E}[p_j]^2}.\end{aligned} \tag{26}$$

Let $\Delta_j \triangleq \mathbb{V}\mathrm{ar}[p_j]/\mathbb{E}[p_j]^2$. Combining (24), (25), and (26), we have

$$\begin{aligned}V^* \le V^{\mathrm{S}} &\le V^* + \frac{1}{2}\sum_{j\in\mathcal{J}} w_j\mathbb{E}[p_j]\left\{\frac{1}{S}\left(2M-1-\Delta_j\right) + \max_{m\in\mathcal{M}}\left[\frac{1}{s_m}\big(\Delta_j - 1\big)\right]\right\} \\ &= V^* + \frac{1}{2}\sum_{j\in\mathcal{J}} w_j\mathbb{E}[p_j]\left\{\left(\frac{2M-1}{S} - \frac{1}{\kappa_j}\right) + \left(\frac{1}{\kappa_j} - \frac{1}{S}\right)\Delta_j\right\},\end{aligned}$$

where $\kappa_j = s_M$ if $\Delta_j > 1$ and $\kappa_j = s_1$ if $\Delta_j \le 1$.

A.5 Proof of Proposition 5.2

We first develop some valid inequalities for all feasible starting times in deterministic scheduling on uniformly related machines; this result is analogous to Lemma 9 in Schulz (1996).

Lemma A.1. *If machines are uniformly related and processing times are deterministic and denoted p_j, then the starting times S_j of jobs for all feasible scheduling policies satisfy, for all $\mathcal{A} \subseteq \mathcal{J}$,*

$$\sum_{j\in\mathcal{A}} p_j S_j \ge -\frac{1}{2}\Big(\frac{1}{s_M} - \frac{1}{S}\Big)\sum_{j\in\mathcal{A}} p_j^2 + \frac{1}{2S}\Big(\sum_{i,j\in\mathcal{A},i\neq j} p_i p_j\Big). \tag{27}$$

Proof. Let $\mathcal{A}_m \subseteq \mathcal{A}$ be the set of jobs processed on machine m. From Schulz et al. (1996), the completion times for all feasible schedules on uniformly related machines satisfy

$$\begin{aligned}\sum_{j\in\mathcal{A}} p_j C_j = \sum_{m=1}^{M} s_m \sum_{j\in\mathcal{A}_m}\frac{p_j}{s_m}C_j &\ge \sum_{m=1}^{M}\frac{s_m}{2}\Big(\sum_{j\in\mathcal{A}_m}\Big(\frac{p_j}{s_m}\Big)^2 + \Big(\sum_{j\in\mathcal{A}_m}\frac{p_j}{s_m}\Big)^2\Big) \\ &= \sum_{m=1}^{M}\frac{1}{2s_m}\Big(\sum_{j\in\mathcal{A}_m} p_j^2 + \Big(\sum_{j\in\mathcal{A}_m} p_j\Big)^2\Big),\end{aligned} \tag{28}$$

where the inequality follows from the single machine case. Since $S_j = C_j - \frac{p_j}{s_m}$ for all $j \in \mathcal{A}_m$,

$$
\begin{aligned}
\sum_{j\in\mathcal{A}} p_j S_j &= \sum_{j\in\mathcal{A}} p_j C_j - \sum_{m=1}^{M} \sum_{j\in\mathcal{A}_m} \frac{p_j^2}{s_m} \\
&\overset{\text{(i)}}{\geq} \sum_{m=1}^{M} \frac{1}{2s_m}\Big(\sum_{j\in\mathcal{A}_m} p_j^2 + \Big(\sum_{j\in\mathcal{A}_m} p_j \Big)^2 \Big) - \sum_{m=1}^{M} \frac{1}{s_m} \sum_{j\in\mathcal{A}_m} p_j^2 \\
&= - \sum_{m=1}^{M} \frac{1}{2s_m} \sum_{j\in\mathcal{A}_m} p_j^2 + \sum_{m=1}^{M} \frac{1}{2s_m}\Big(\sum_{j\in\mathcal{A}_m} p_j \Big)^2 \\
&\overset{\text{(ii)}}{\geq} - \sum_{m=1}^{M} \frac{1}{2s_m} \sum_{j\in\mathcal{A}_m} p_j^2 + \frac{1}{2S}\Big(\sum_{j\in\mathcal{A}} p_j \Big)^2 \\
&\geq -\frac{1}{2s_M} \sum_{j\in\mathcal{A}} p_j^2 + \frac{1}{2S}\Big(\sum_{j\in\mathcal{A}} p_j \Big)^2 \\
&= -\frac{1}{2}\Big(\frac{1}{s_M} - \frac{1}{S} \Big) \sum_{j\in\mathcal{A}} p_j^2 + \frac{1}{2S}\Big(\sum_{i,j\in\mathcal{A}, i\neq j} p_i p_j \Big),
\end{aligned}
$$

where (i) is due to (28) and (ii) follows from the inequality $\sum_{m=1}^{M} \frac{x_m^2}{s_m} \geq \frac{(\sum_{m=1}^{M} x_m)^2}{\sum_{m=1}^{M} s_m}$ for any non-negative numbers $\{x_m\}_{m=1}^{M}$ and $\{s_m\}_{m=1}^{M}$, which can be proved easily by induction. □

Using Lemma A.1, we now show Proposition 5.2. Since p_j and S_j are independent for any non-anticipative policy, and p_j and p_i are independent for all $i \neq j$, taking expectations on both sides of (27) gives

$$
\begin{aligned}
\sum_{j\in\mathcal{A}} \mathbb{E}[p_j]\mathbb{E}[S_j] &\geq -\frac{1}{2}\Big(\frac{1}{s_M} - \frac{1}{S} \Big) \sum_{j\in\mathcal{A}} \mathbb{E}[p_j^2] + \frac{1}{2S}\Big(\sum_{i,j\in\mathcal{A}, i\neq j} \mathbb{E}[p_i]\mathbb{E}[p_j] \Big) \\
&= \frac{1}{2S}\Big(\sum_{j\in\mathcal{A}} \mathbb{E}[p_j] \Big)^2 - \frac{1}{2}\Big(\frac{1}{s_M} - \frac{1}{S} \Big) \sum_{j\in\mathcal{A}} \mathbb{V}\mathrm{ar}[p_j] - \frac{1}{2s_M} \sum_{j\in\mathcal{A}} \mathbb{E}[p_j]^2.
\end{aligned}
\tag{29}
$$

Using the fact that $\mathbb{E}[C_j] \geq \mathbb{E}[S_j] + \frac{\mathbb{E}[p_j]}{s_1}$ and (29), we obtain (7).

B Additional Proofs and Derivations

B.1 Performance Bound on Static Routing With Dependent Jobs

Theorem B.1 (Extension of Theorem 4.1). *If $\frac{\mathbb{V}\mathrm{ar}[p_{jm}]}{\mathbb{E}[p_{jm}]^2} \le \Delta$ holds for all jobs $j \in \mathcal{J}$ and machines $m \in \mathcal{M}$, then any static routing policy obtained from an optimal solution of (2) satisfies*

$$\begin{aligned} V^* \le V^{\mathrm{R}} \le V^* &+ \Big(\frac{M-1}{2M} + \frac{\Delta}{2}\Big) \sum_{j\in\mathcal{J}} w_j \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}] + \sum_{j\in\mathcal{J}} \alpha_j w_j \\ &+ \Big(2+\sqrt{\Delta M}\Big)\Big(\sum_{j\in\mathcal{J}} \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}]\Big)\Big(\sum_{j\in\mathcal{J}} \alpha_j \max_{m\in\mathcal{M}} r_{jm}\Big) + \frac{1}{2}\sum_{j\in\mathcal{J}} \beta_j \max_{m\in\mathcal{M}} r_{jm}. \end{aligned} \tag{1}$$

Proof. In the proof we use similar penalties as in Section 4.2 with the same parameters λ_{jm} and γ_{jm} as in Proposition 4.5. We first state and prove Lemma B.2, which bounds the parameters λ_{jm} in the routing penalty.

Lemma B.2. *Let $\lambda_{jm} = \nu_j^*/\mathbb{E}[p_{jm}] + \frac{1}{2}w_j$ with ν_j^* being an optimal Lagrange multiplier of the assignment constraint $\sum_{m\in\mathcal{M}} x_{jm} = 1$ in (2). We have the following inequalities*

$$0 \le \lambda_{jm} \le w_j + r_{jm} \sum_{i\in\mathcal{J}} \mathbb{E}[p_{im}]\,. \tag{2}$$

Proof. It suffices to show any optimal Lagrange multiplier ν_j^* of the assignment constraint $\sum_{m\in\mathcal{M}} x_{jm} = 1$ satisfies

$$0 \le \nu_j^* \le \frac{1}{2} w_j \mathbb{E}[p_{jm}] + r_{jm}\mathbb{E}[p_{jm}] \sum_{i\in\mathcal{J}} \mathbb{E}[p_{im}], \quad \forall m \in \mathcal{M}.$$

To show this, let $\mathbf{x}^* = (x^*_{jm})_{j\in\mathcal{J}, m\in\mathcal{M}} \in \mathbb{R}^{J\times M}$ be an optimal solution and $\mu^*_{jm} \ge 0$ be an optimal Lagrange multiplier of the non-negativity constraint $x_{jm} \ge 0$ in (2). By the KKT conditions, for all jobs $j \in \mathcal{J}$ and machines $m \in \mathcal{M}$ we have

$$\frac{1}{2} w_j \mathbb{E}[p_{jm}] + \sum_{i\in\mathcal{J}} d^m_{ji} x^*_{im} = \nu^*_j + \mu^*_{jm}, \tag{3}$$

where $d^m_{ji} = \min(r_{im}, r_{jm})\mathbb{E}[p_{jm}]\mathbb{E}[p_{im}]$ is the (j,i)-th element in $\mathbf{D}_m$ as is defined in Lemma 3.1.

We first show $\nu^*_j \ge 0$. If not, from (3) we have $\mu^*_{jm} > 0$ for all $m \in \mathcal{M}$. Then according to the complementary slackness we have $x^*_{jm} = 0$ for all $m \in \mathcal{M}$. This contradicts the assignment constraint, which requires $\sum_{m\in\mathcal{M}} x^*_{jm} = 1$.

Next, since $\mu^*_{jm} \ge 0$ for all $m \in \mathcal{M}$, from (3) we have for all $m \in \mathcal{M}$,

$$\nu^*_j \le \frac{1}{2} w_j \mathbb{E}[p_{jm}] + \sum_{i\in\mathcal{J}} d^m_{ji} x^*_{im} \le \frac{1}{2} w_j \mathbb{E}[p_{jm}] + r_{jm}\mathbb{E}[p_{jm}] \sum_{i\in\mathcal{J}} \mathbb{E}[p_{im}]\,,$$

because $x^*_{im} \le 1$ and $d^m_{ji} \le r_{jm}\mathbb{E}[p_{jm}]\mathbb{E}[p_{im}]$. The result follows. □

We now prove Theorem B.1. We will restrict attention to non-anticipative policies where idling all machines at the same time is not allowed. This idling constraint is trivially satisfied by the optimal non-anticipative policy, and it allows us to bound the starting times of each job in the perfect information problem.

First, note that for the upper bound on V^{R}, Proposition 4.3 still holds. To derive a lower bound on V^* using a penalized perfect information relaxation, we define the sequencing penalty Y_{S} to be

$$Y_{\mathrm{S}} \triangleq \sum_{j\in\mathcal{J}} \sum_{m\in\mathcal{M}} r_{jm} S_{jm}^{\pi}\big(p_{jm} - \mathbb{E}[p_{jm}|\mathcal{F}_{S_{jm}^{\pi}}^{\pi}]\big), \tag{4}$$

and the routing penalty Y_{R} to be

$$Y_{\mathrm{R}} \triangleq \sum_{j\in\mathcal{J}} \sum_{m\in\mathcal{M}} x_{jm}^{\pi}\Big[\lambda_{jm}\big(\mathbb{E}[p_{jm}|\mathcal{F}_{S_{jm}^{\pi}}^{\pi}] - p_{jm}\big) + \gamma_{jm}\big(\mathbb{E}[p_{jm}^2|\mathcal{F}_{S_{jm}^{\pi}}^{\pi}] - p_{jm}^2\big)\Big]. \tag{5}$$

We set the parameters $(\lambda_{jm})_{j\in\mathcal{J},m\in\mathcal{M}}$ and $(\gamma_{jm})_{j\in\mathcal{J},m\in\mathcal{M}}$ as in the statement of Proposition 4.5. Since S_{jm}^{π} is measurable with respect to $\mathcal{F}_{S_{jm}^{\pi}}^{\pi}$ for all non-anticipative policies, we obtain by the law of total expectation that the sequencing penalty Y_{S} is dual feasible. An analogous argument shows the routing penalty Y_{R} is also dual feasible, as x_{jm}^{π} is measurable with respect to $\mathcal{F}_{S_{jm}^{\pi}}^{\pi}$ for all non-anticipative policies. Let

$$V^{\mathrm{H}}(\mathbf{p}) = \min_{\mathbf{x}\in\mathcal{X}}\Big\{\sum_{j\in\mathcal{J}} w_j C_j + Y_{\mathrm{S}} + Y_{\mathrm{R}}\Big\},$$

be the penalized perfect information relaxation problem for sample path $\mathbf{p} \in \mathbb{R}^{J\times M}$. We have

$$V^{\mathrm{H}} \triangleq \mathbb{E}[V^{\mathrm{H}}(\mathbf{p})] \le V^*, \tag{6}$$

is a lower bound on V^*, because the penalty $Y_{\mathrm{S}} + Y_{\mathrm{R}}$ is dual feasible.

Next, define the penalties as in Section 4:

$$\tilde{Y}_{\mathrm{S}} \triangleq \sum_{j\in\mathcal{J}} \sum_{m\in\mathcal{M}} r_{jm} S_{jm}^{\pi}\big(p_{jm} - \mathbb{E}[p_{jm}]\big),$$

$$\tilde{Y}_{\mathrm{R}} \triangleq \sum_{j\in\mathcal{J}} \sum_{m\in\mathcal{M}} x_{jm}^{\pi}\Big[\lambda_{jm}\big(\mathbb{E}[p_{jm}] - p_{jm}\big) + \gamma_{jm}\big(\mathbb{E}[p_{jm}^2] - p_{jm}^2\big)\Big],$$

and let $\tilde{V}^{\mathrm{H}}(\mathbf{p})$ be the perfect information problem with these penalties:

$$\tilde{V}^{\mathrm{H}}(\mathbf{p}) = \min_{\mathbf{x}\in\mathcal{X}}\Big\{\sum_{j\in\mathcal{J}} w_j C_j + \tilde{Y}_{\mathrm{S}} + \tilde{Y}_{\mathrm{R}}\Big\}.$$

We have that

$$V^{\mathrm{H}}(\mathbf{p}) \ge \tilde{V}^{\mathrm{H}}(\mathbf{p}) + \min_{\mathbf{x}\in\mathcal{X}}\Big\{Y_{\mathrm{S}} + Y_{\mathrm{R}} - \tilde{Y}_{\mathrm{S}} - \tilde{Y}_{\mathrm{R}}\Big\}. \tag{7}$$

Using the assumption of limited dependence (10) and (11), we can bound the difference in penalties as follows:

$$Y_{\mathrm{S}} + Y_{\mathrm{R}} - \tilde{Y}_{\mathrm{S}} - \tilde{Y}_{\mathrm{R}} \ge -\sum_{j\in\mathcal{J}} \alpha_j \sum_{m\in\mathcal{M}} r_{jm} S_{jm} - \sum_{j\in\mathcal{J}} \alpha_j \sum_{m\in\mathcal{M}} x_{jm}|\lambda_{jm}| - \sum_{j\in\mathcal{J}} \beta_j \sum_{m\in\mathcal{M}} x_{jm}|\gamma_{jm}|.$$

We next bound each of these terms. For the first term, we use the fact that the start time of every job $j \in \mathcal{J}$ must satisfy $S_j \le C_j \le \sum_{i\in\mathcal{J}} \max_{m\in\mathcal{M}} p_{im}$ because idling all machines at the same time is not allowed. Moreover, because each job is processed by one machine we have that

$\sum_{m\in\mathcal{M}} r_{jm}S_{jm} \le \big(\max_{m\in\mathcal{M}} r_{jm}\big)S_j$. Therefore,

$$\sum_{j\in\mathcal{J}} \alpha_j \sum_{m\in\mathcal{M}} r_{jm}S_{jm} \le \Big(\sum_{j\in\mathcal{J}} \max_{m\in\mathcal{M}} p_{jm}\Big)\Big(\sum_{j\in\mathcal{J}} \alpha_j \max_{m\in\mathcal{M}} r_{jm}\Big).$$

For the second term, the assignment constraint and Lemma B.2 imply that

$$\sum_{j\in\mathcal{J}} \alpha_j \sum_{m\in\mathcal{M}} x_{jm}|\lambda_{jm}| \le \sum_{j\in\mathcal{J}} \alpha_j w_j + \Big(\sum_{j\in\mathcal{J}} \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}]\Big)\Big(\sum_{j\in\mathcal{J}} \alpha_j \max_{m\in\mathcal{M}} r_{jm}\Big).$$

For the third term, we use $\gamma_{jm} = -\frac{1}{2}r_{jm}$ and the assignment constraint to obtain that

$$\sum_{j\in\mathcal{J}} \beta_j \sum_{m\in\mathcal{M}} x_{jm}|\gamma_{jm}| \le \frac{1}{2}\sum_{j\in\mathcal{J}} \beta_j \max_{m\in\mathcal{M}} r_{jm}.$$

Taking expectations on both sides of (7) gives

$$\begin{aligned} V^{\mathrm{H}} \ge \mathbb{E}\left[\tilde{V}^{\mathrm{H}}(\mathbf{p})\right] - \sum_{j\in\mathcal{J}} \alpha_j w_j - \left(2+\sqrt{\Delta M}\right)\Big(\sum_{j\in\mathcal{J}} \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}]\Big)\Big(\sum_{j\in\mathcal{J}} \alpha_j \max_{m\in\mathcal{M}} r_{jm}\Big) \\ - \frac{1}{2}\sum_{j\in\mathcal{J}} \beta_j \max_{m\in\mathcal{M}} r_{jm}, \end{aligned} \tag{8}$$

because

$$\mathbb{E}\Big[\max_{m\in\mathcal{M}} p_{jm}\Big] \le \max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}] + M^{1/2}\max_{m\in\mathcal{M}} \left(\mathbb{V}\mathrm{ar}[p_{jm}]\right)^{1/2} \le \left(1+\sqrt{\Delta M}\right)\max_{m\in\mathcal{M}} \mathbb{E}[p_{jm}],$$

due to Devroye (1979) and $\mathbb{V}\mathrm{ar}[p_{jm}] \le \Delta\mathbb{E}[p_{jm}]^2$ for all jobs $j \in \mathcal{J}$ and machines $m \in \mathcal{M}$. The result follows from using Proposition 4.3 to bound V^{R} from above, Proposition 4.5 to bound $\tilde{V}^{\mathrm{H}}(\mathbf{p})$ from below, (6), and (8). □

B.2 Additional Details on Example 6.1

Because weights are one, we can express V^{A} as

$$V^{\mathrm{A}} = \mathbb{E}\Bigg[\underbrace{\sum_{i=1}^{J/2} C_{2i-1}}_{(a)} + \underbrace{\sum_{i=1}^{J/2} C_{2i}\mathbf{1}_{\{p_{2i-1}=0\}}}_{(b)} + \underbrace{\sum_{i=1}^{J/2} C_{2i}\mathbf{1}_{\{p_{2i-1}=1\}}}_{(c)}\Bigg],$$

where (a) represents the completion time of odd-numbered jobs, (b) represents the completion time of even-numbered jobs processed immediately, and (c) represents the completion time of even-numbered jobs delayed to the end of sequence. The odd-numbered job $2i-1$ has to wait for all jobs with $j < 2i-1$ to either complete processing on the machine or transfer towards the end of the sequence. Let t_i be the total processing time of job pair (p_{2i-1}, p_{2i}) excluding the time to process job $2i$ if delayed. We have

$$\mathbb{E}[t_i] = \mathbb{E}[p_{2i-1}] + \mathbb{E}[p_{2i}\mathbf{1}\{p_{2i-1}=0\}] = \frac{1}{2}\cdot 1 + \frac{1}{2}\cdot(1-2q_i) = 1-q_i.$$

Thus,

$$\mathbb{E}[(a)] = \sum_{i=1}^{J/2}\Big(\mathbb{E}[p_{2i-1}] + \sum_{j<i}\mathbb{E}[t_j]\Big) = \sum_{i=1}^{J/2}\Big(\frac{1}{2} + \sum_{j<i}(1-q_j)\Big) = \frac{1}{8}J^2 - \sum_{i=1}^{\frac{J}{2}}\sum_{j<i} q_j.$$

Similar, we have

$$\begin{aligned}\mathbb{E}[(b)] &= \sum_{i=1}^{J/2}\mathbb{P}(p_{2i-1}=0)\Big(\mathbb{E}[p_{2i}|p_{2i-1}=0] + \sum_{j<i}\mathbb{E}[t_j]\Big) = \sum_{i=1}^{J/2}\frac{1}{2}\Big((1-2q_i) + \sum_{j<i}(1-q_j)\Big)\\ &= \frac{1}{16}J^2 + \frac{1}{8}J - \frac{1}{2}\sum_{i=1}^{\frac{J}{2}}\sum_{j<i} q_j - \sum_{i=1}^{\frac{J}{2}} q_i.\end{aligned}$$

To calculate (c), we first calculate the expectation conditioning on the realized processing times of all odd-numbered jobs $\{p_{2i-1}\}_{i=1}^{\frac{J}{2}}$, then we take expectation over these processing times. We let $N = \sum_{i=1}^{\frac{J}{2}} p_{2i-1}$ be the number of odd-numbered jobs with realized processing times of one. Altogether N even-numbered jobs will be moved towards the end of the sequence, because job $2i$ is delayed if $p_{2i-1}=1$. Because the machine begins processing these jobs at time $N + \sum_{i=1}^{\frac{J}{2}}(1-p_{2i-1})\mathbb{E}[p_{2i}|p_{2i-1}=0]$, we have that

$$\begin{aligned}\mathbb{E}\left[(c) \mid \{p_{2i-1}\}_{i=1}^{\frac{J}{2}}\right] &= N\Big(N + \sum_{i=1}^{\frac{J}{2}}(1-p_{2i-1})\mathbb{E}[p_{2i}|p_{2i-1}=0]\Big) + \sum_{i=1}^{\frac{J}{2}}\sum_{j\le i} p_{2i-1}p_{2j-1}\mathbb{E}[p_{2j}|p_{2j-1}=1]\\ &= N\Big(N + \sum_{i=1}^{\frac{J}{2}}(1-p_{2i-1})(1-2q_i)\Big) + \sum_{i=1}^{\frac{J}{2}}\sum_{j\le i} p_{2i-1}p_{2j-1}2q_j.\end{aligned}$$

Thus, we obtain by taking expectations over $\{p_{2i-1}\}_{i=1}^{\frac{J}{2}}$

$$\begin{aligned}\mathbb{E}[(c)] &= \frac{1}{8}J^2 - \big(\frac{J}{2}-2\big)\sum_{i=1}^{\frac{J}{2}} q_i + \sum_{i=1}^{\frac{J}{2}}\sum_{j<i} q_j + \frac{1}{2}\sum_{i=1}^{\frac{J}{2}}\sum_{j>i} q_j\\ &\le \frac{1}{8}J^2 - \big(\frac{J}{2}-2\big)\sum_{i=1}^{\frac{J}{2}} q_i + \frac{3}{2}\sum_{i=1}^{\frac{J}{2}}\sum_{j<i} q_j,\end{aligned}$$

where the inequality is because q_i are sequenced in decreasing order.

Combining (a), (b), and (c) yields

$$V^{\mathrm{A}} \le \frac{5}{16}J^2 + \frac{1}{8}J + \big(1 - \frac{J}{2}\big)\sum_{i=1}^{\frac{J}{2}} q_i.$$

C Dynamic Programming Formulation

The optimal policy of the scheduling problem described in Section 2 can be obtained by solving Bellman's Equation, using backward induction. For sake of simplicity of presentation, we assume all random variables p_{jm} take only positive integral values, that is, $\mathbb{P}\{p_{jm} \in \mathbb{N}\} = 1$. Then we can further assume without loss of generality that jobs can only be started at integral points in time $t \in \mathbb{N}$.

We define the state to be (W, P, ℓ), where $W \subseteq \mathcal{J}$ is the subset of jobs waiting for service, and $P \subseteq \mathcal{J} \times \mathcal{M}$ is the subset of job-machine pairs under process. $\ell = (\ell_j)_{j \in \mathcal{J}} \in \mathbb{R}^J$ is defined as follows: if job j is currently under process on some machine, then ℓ_j is the elapsed processing time of job j; if job j is waiting for service, then $\ell_j = 0$; otherwise if job j is completed, ℓ_j can take an arbitrary value.

To describe Bellman's Equation, we introduce some helpful notation. Suppose $P \subseteq \mathcal{J} \times \mathcal{M}$ is a subset of job-machine pairs. We define two operations $\mathcal{J}(\cdot)$ and $\mathcal{M}(\cdot)$ on P such that $\mathcal{J}(P)$ and $\mathcal{M}(P)$ give the jobs and machines under process in the job-machine pairs in P, respectively. Also, for each set $A \subseteq \mathcal{J}$, we define the vector $\mathbf{I}_A = (\mathrm{I}_A^j)_{j \in \mathcal{J}} \in \{0,1\}^J$ to be $\mathrm{I}_A^j = 1$ if $j \in A$, and $\mathrm{I}_A^j = 0$ otherwise.

Let $V(W, P, \ell)$ be the cost-to-go function when the state is (W, P, ℓ). If $W = \varnothing$, set $V(W, P, \ell) = \sum_{(j,m) \in P} w_j \mathbb{E}\left[p_{jm} - l_j | p_{jm} > l_j\right]$. If $W \neq \varnothing$, then the action set is

$$\mathcal{A}(W, P) = \{\hat{P} \subseteq W \times \mathcal{M} \setminus \mathcal{M}(P) : \text{if } (j, m), (j', m') \in \hat{P}, \text{then } j \neq j' \text{and } m \neq m'\}.$$

That is, $\mathcal{A}(W, P)$ is the set of all possible job-machine pairs that can be assigned when jobs in W are waiting and job-machine pairs in P are under process. Thus, if $\hat{P} \in \mathcal{A}(W, P)$ is chosen, the decision maker will assign job j to currently idle machine m for each job-machine pair $(j, m) \in \hat{P}$.

In addition, define $\mathcal{F} \subseteq P$ to be the random set of job-machine pairs that are currently under process and will be completed by the beginning of next stage, i.e., for each set $F \subseteq P$,

$$\mathbb{P}\{\mathcal{F} = F | P, \ell\} = \prod_{(j,m) \in P \setminus F} \mathbb{P}\{p_{jm} > \ell_j + 1 | p_{jm} > \ell_j\} \prod_{(j,m) \in F} \mathbb{P}\{p_{jm} = \ell_j + 1 | p_{jm} > \ell_j\}.$$

Then the Bellman Equation can be written as:

$$V(W, P, \ell) = \min_{\hat{P} \in \mathcal{A}(W,P)} \left\{ \sum\nolimits_{j \in \mathcal{J}(P) \cup W} w_j + \mathbb{E}_{\mathcal{F} | P \cup \hat{P}, \ell} \left[V\left(W \setminus \mathcal{J}(\hat{P}), \{P \cup \hat{P}\} \setminus \mathcal{F}, \ell + \mathbf{I}_{\mathcal{J}\left(\{P \cup \hat{P}\} \setminus \mathcal{F}\right)} \right) \right] \right\}.$$

The optimal performance is given by $V^* = V(\mathcal{J}, \varnothing, \mathbf{0})$.

D Binary Linear Programming Formulation for Optimal Static Routing

In this section, we provide an approach to calculating an exact solution to optimization problem (1) using 0-1 integer programming. The approach may be viable when the number of jobs is not large.

We can write problem (1) in matrix form as

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{m \in \mathcal{M}} \mathbf{c}'_m \mathbf{x}_m + \frac{1}{2} \mathbf{x}'_m \tilde{\mathbf{D}}_m \mathbf{x}_m, \tag{9}$$

where $\mathbf{c}_m = (w_j \mathbb{E}[p_{jm}])_{j \in \mathcal{J}} \in \mathbb{R}^J$, and $\tilde{\mathbf{D}}_m = (\tilde{d}_{ij}^m) \in \mathbb{R}^{J \times J}$ is the matrix such that

$$\tilde{d}_{ij}^m = \begin{cases} 0 & i = j, \\ w_i \mathbb{E}[p_{jm}] & j \prec_m i, \\ w_j \mathbb{E}[p_{im}] & i \prec_m j. \end{cases}$$

We claim that the optimal value of problem (9) does not change if we replace the non-negative constraint $x_{jm} \geq 0$ with the binary constraint $x_{jm} \in \{0, 1\}$ for each job-machine pair. To see this, suppose $\mathbf{x}$ is an optimal solution for (9), then using the method of conditional probabilities we can derandomize $\mathbf{x}$ to a feasible binary solution whose performance cannot be any worse. On the other hand, every feasible binary solution can not be any better than $\mathbf{x}$, because $\mathbf{x}$ is an optimal solution to (9).

Define new binary variables $y_{ij}^m \triangleq x_{im} x_{jm}$. Then the objective of problem (9) is equivalent to

$$\sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} c_{jm} x_{jm} + \sum_{m \in \mathcal{M}} \sum_{i < j \in \mathcal{J}} y_{ij}^m \tilde{d}_{ij}^m.$$

We can guarantee that $y_{ij}^m = x_{im} x_{jm}$ using the following constraints:

$$\begin{aligned} & y_{ij}^m \leq x_{jm}, \\ & y_{ij}^m \leq x_{im}, \\ & y_{ij}^m \geq x_{jm} + x_{im} - 1, \\ & y_{ij}^m \in \{0, 1\}. \end{aligned}$$

As a result, problem (9) is equivalent to

$$\begin{aligned} \underset{x_{jm}, y_{ij}^m}{\text{minimize}} \quad & \sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{J}} c_{jm} x_{jm} + \sum_{m \in \mathcal{M}} \sum_{i < j \in \mathcal{J}} y_{ij}^m \tilde{d}_{ij}^m \\ \text{subject to} \quad & \sum_{m \in \mathcal{M}} \mathbf{x}_m = \mathbf{1}, \\ & y_{ij}^m \leq x_{jm}, \\ & y_{ij}^m \leq x_{im}, \\ & y_{ij}^m \geq x_{jm} + x_{im} - 1, \\ & x_{jm} \in \{0, 1\}, \\ & y_{ij}^m \in \{0, 1\}, \end{aligned}$$

which is a 0-1 integer programming problem.

References

Devroye, L. P. (1979), 'Inequalities for the completion times of stochastic pert networks', *Mathematics of Operations Research* **4**(4), 441–447.