A Model for Make-To-Order Revenue Management[1]

Jérémie Gallien[2], Yann Le Tallec[3], Tor Schoenmeyr[4]

November 18, 2004

Abstract

Seeking to help practitioners establish quantitative guidelines for negotiating make-to-order contracts along the dimensions of price, quantity and lead-time, we investigate the dynamic admission control of jobs with hard deadlines into a single machine queue with preemptive scheduling. Using the concept of minimum workload function, we establish that earliest due-date scheduling can be assumed at no cost to optimality, and propose a discrete-time formulation for the problem of maximizing long-run expected profit. We establish some properties and a characterization of the optimal policy, which we exploit to derive two heuristic policies (*fluid* and *lookahead*) relying on different approximations for the opportunity cost of accepting a job. Numerical experiments under various load, stretch and granularity parameters suggest that they always perform better than common simple static policies. Limited experiments also suggest that the optimal static policy may perform nearly as well as our two dynamic heuristics. While that policy is simple to implement however, it seems challenging to derive using known methods. Overall, our fluid heuristic stands out for its robust performance at a relatively low computational cost, and its possible extensions in practice to non-stationary demand and orders with staggered deliveries.

1. Introduction

1.1. Motivation. Many firms operate in a make-to-order fashion, either because the product they sell is unique to each customer (e.g. print shop, laundry service, commercial DNA sequencing), or because they seek to offer greater product variety at a lower cost (e.g. consumer electronics and PC assembly). Their customers are frequently price-sensitive and/or time-sensitive, so that quoted prices and lead-times typically impact benefits and market share significantly. The present paper describes a quantitative model designed to improve the function known as *order promising* in make-to-order firms facing capacity constraints. That is, we seek to help the salesforce of those firms realize the full profit potential associated with the production capacity available when quoting prices and lead-times to prospective customers.

For modeling purposes, we assume that the terms of price, lead-time and quantity (or

 Feedback on this paper is welcome and appreciated.

[2] Corresponding author: MIT Sloan School of Management, Cambridge, MA 02142, jgallien@mit.edu

[3] MIT Operations Research Center, letallec@mit.edu

[4] MIT Sloan School of Management, tor_s@mit.edu

capacity requirement) entirely characterize each transaction. In particular, other possible differentiating terms (e.g. payment delays, warranty, insurance, return policy) are ignored for now, as is the important notion of a "strategic" customer that could be offered advantageous sales conditions in order to build a long-term relationship or to generate publicity (we will come back to those issues later in §5). Under this assumption, we propose that any consistent salesforce guidelines for a firm can be represented at each point in time by an *acceptance region* (the subset of all transactions deemed profitable then) in the 3-dimensional space of price, quantity and lead-time. We naturally define its complement as the *rejection region*, and the border between those two regions as the *admission surface*. Our goal is to derive an optimal admission control policy, that is develop a methodology for dynamically computing admission surfaces as a function of the supply capacity available in order to maximize profits in the long run.

Note that our use of the terms "admission" and "rejection" above suggest a situation where customers request transaction terms, and the firm's only control is to either accept of reject them (i.e. the firm is *terms taker*). In contrast, sales agents often take a substantially more active role in practice, and typically make price and lead-time quotes after a customer describes product type(s) and quantity desired. In addition, the actual interaction between a sales agent and a prospective customer is frequently an iterative negotiation, with both parties exchanging offers and counter-offers. The quantitative model presented in this paper does assume that the firm *is* terms taker, and the admission surfaces we eventually compute are theoretically only valid under that restrictive assumption. However, we are motivated by the belief that these admission surfaces may still allow to determine useful salesforce guidelines in situations also involving active quotations and negotiations, even if we do not provide any theoretical justification for such practice. For example, a sales agent asked to provide a quote for a given quantity of some product type under a specified lead-time may use the price coordinate (or some proportionately larger value) of the corresponding point on the admission surface. Also, any feasible but unprofitable transaction requested by a customer corresponds to a point in the rejection region which may be projected onto the admission surface: if such a request is made by a price-sensitive customer, the lead-time coordinate of that projection along the axis of lead-time onto the surface would arguably suggest a sensible counter-offer; Likewise, the price coordinate of the projection along the price axis may suggest a sensible counter-offer if the sales agent believes instead that the customer is more time-sensitive – see Figure 1 for an example. In summary, we thus believe our work to be potentially helpful for negotiation and active quotations as well, even if our theoretical model exclusively focuses on admission control and does not attempt to capture negotiation dynamics and the utility function of individual customers.

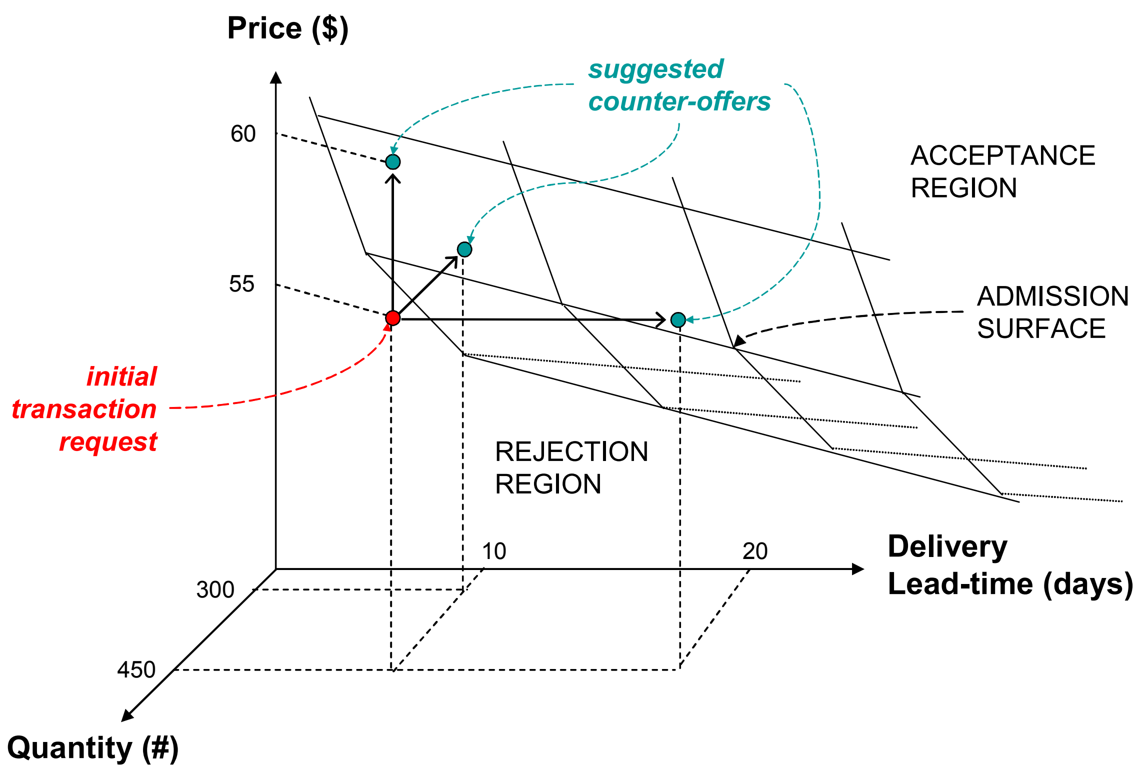

Figure 1: The admission surface and negotiation geometry

Intuition suggests that the optimal acceptance region should evolve dynamically (e.g. expanding when there is much idle capacity and shrinking in the opposite case), and that it should depend on the forecast of future orders. Indeed, the key challenge is to determine the opportunity cost of using some capacity for a given incoming order, so it can be compared with the profit obtained from accepting it. This feature is highly reminiscent of the problem known in the field of airline revenue management as inventory control (allocation over time of remaining flight seats to various fare classes, see van Ryzin and Talluri 2004), which justifies our title. As will be seen from our model description in §2.1, another resemblance with many airline revenue management studies is our modeling approach, which consists of discretizing the space of possible customers into a finite set of demand classes. But we also find it insightful to point out some differences between our problem and airline seat inventory control:

(i) Production capacity is allocated over an infinite horizon, as opposed to the finite seat allocation period ending with flight departure;

(ii) Capacity remaining at any point in time is characterized by the amount of production available by all possible future due-dates, as opposed to a number of seats left;

(iii) Consumption of production capacity by customers is continuous, and its timing (order scheduling) constitutes a decision variable constrained by different customer lead-times. In contrast, "consumption" of seat capacity (flight departure) is instantaneous, exogenous and simultaneous for all customers.

In the rest of this paper, we discuss the related literature in §1.2, and present our mathematical model formulation in §2, which includes a problem statement in §2.1 and a dynamic programming formulation in §2.2. We begin our analysis section §3 by presenting some properties of the optimal policy in §3.1, then describe our proposed heuristic policies in §3.2. We report the results of experiments designed to assess their relative performance in §4, and Section §5 concludes the paper with a summary of results and a discussion of implementation issues. All proofs are relegated to an appendix.

1.2. Literature Review. In line with the main features of our model to be described in §2, we mostly restrict our discussion to papers jointly investigating the optimal admission control (possibly through due-date or price quotations) and scheduling of several job classes for a single make-to-order server.

A first such stream of papers originates in the branch of the computer science literature known as online scheduling, typically assumes (as we do) deterministic service times with hard due-date constraints, but (differing from our work) considers the objective of minimizing worst-case performance through the concept of competitive ratio. That is, no assumptions are made about the arrival process of future incoming jobs, and relatively little if anything is typically assumed about their characteristics. The objective pursued is to design an online admission control and scheduling policy with a guaranteed performance as close as possible to that of a *clairvoyant* scheduler (i.e. with perfect knowledge of future jobs) over all possible instances of job arrival sequences. The most relevant such papers include Locke (1986) and Koren and Shasha (1995) for the preemptive case, Goldman et al. (2000) and Goldwasser (2003) for the non-preemptive case.

A second set of papers, found in the operations management literature, is closer to our work in that it considers (as we do) the objective of maximizing long-run average profit, and also assumes some probabilistic structure for the arrival of future jobs. Differing from our model however, these papers assume stochastic service times; Duenyas (1995) considers a model where customers arrive according to Poisson processes, and the probability that they accept a given order is a class-dependent function of the quoted due-date. Saliently, all classes share the same service time distribution and linear tardiness cost. In that setting, Duenyas shows that the optimal sequencing rule is EDD (earliest due-date), which we also find to be true in our model, and proposes a heuristic for admission control based on the approximation that the sequencing rule used is $FCFS$ (first-come-first-serve). He also argues that a joint admission control and sequencing model may be helpful in practice when jointly negotiating price and lead-times with customers, a primary motivation for the present paper.

The model of Plambeck et al. (2001) involves job classes with renewal input processes and general service time distributions as well as class-dependent rejection penalties and upper bound constraints on throughput time. The admission control policy they investigate consists of rejecting jobs from the class with the lowest penalty per unit of capacity required when the total workload is above some threshold, and always accepting jobs from the other classes. Their scheduling policy dynamically allocates machine capacity to the job class with the highest relative backlog, defined for each class as the ratio of the number of queued jobs from that class over a nominal number of jobs equal to its arrival rate times its throughput time upper bound. Plambeck et al. show that, in the heavy-traffic regime, the policy just mentioned complies asymptotically with the throughput time upper bound constraints, and is also asymptotically optimal.

Plambeck (2004) studies the problem of setting static prices and dynamically quoting lead-times for two customer classes (price-sensitive and time-sensitive customers); arrivals follow nonhomogeneous Poisson processes with class-dependent rates decreasing linearly with price and lead-times, but the two classes share the same exponential service time distribution. She derives through an asymptotic analysis a policy consisting of giving scheduling priority and promising immediate delivery to time-sensitive customers, and quoting price-sensitive customers a lead-time proportional to the current workload. Using a methodology reminiscent of Plambeck et al. (2001), she establishes that the policy described complies asymptotically with the quoted lead-times, and is also asymptotically optimal in heavy traffic.

In a recent paper, Maglaras (2003) considers a model where admission control is effectively performed through dynamic pricing decisions impacting a specified general demand model, job arrivals from the different classes follow Poisson processes, and accepted orders have class-dependent quadratic holding costs and exponential service times. Combining diffusion and fluid approximations, Maglaras derives a pricing policy only depending on system workload, which consists of maximizing the instantaneous average revenue rate under a resource utilization target constraint effectively stabilizing the system in the heavy-traffic regime; his proposed scheduling policy is the generalized $c\mu$ rule described in Van Mieghem (1995).

Finally, Kapuscinski and Tayur (2003) study a single server quoting due-dates entailing class-dependent waiting costs for two classes of customers arriving over time. While that model differs in some important aspects from both the present paper and the others just quoted above (no admission control, discrete-time, finite horizon), it also bears some resemblance to our model (stochastic arrivals but deterministic processing times and hard due-date constraints). Noticeably, Kapuscinski and Tayur also consider the function of remaining slack over time for a given deterministic schedule, which plays a fundamental role in our analysis.

2. Problem Definition

The approach we develop for dynamically computing the admission surface described in §1.1 relies on a discretization of the transaction space. Specifically, we describe next a dynamic admission control model conceptually allowing to determine at any given time whether each one of a large but finite number of transactions (the job classes defining the discretization mentioned above) should belong to the admission or rejection region – the admission surface may then be obtained as the border between two sets of discrete points. While the remainder of this section as well as Sections §3 and §4 are dedicated to this admission control subproblem (i.e. dynamically deciding whether any given incoming job should be accepted or rejected), Section §5 contains a discussion of implementation issues arising when using the admission control policies we eventually obtain in order to generate admission surfaces over time.

2.1. Model Description. Our model describes a production facility with limited capacity facing a random arrival stream of transaction opportunities with various capacity requirements, due-dates and profits. The objective is to find admission control and scheduling policies maximizing the expected profit rate in the long run, while satisfying the due-date constraints of all accepted jobs.

We assume that each incoming transaction opportunity (or job) belongs to one of J job classes (the set of all job classes is denoted $\mathbb{J}$), where each class $j \in \mathbb{J}$ is characterized by four deterministic quantities including a *processing time* q_j, a *slack time* s_j, a *profit* r_j, and an *arrival rate* λ_j:

- The processing time q_j of a job from class j is the time that would be required to complete it from start to finish if the production facility were solely dedicated to it. Note that this definition only depends on the amount of capacity required by a given transaction, so that our model allows for possibly many product types. Once a job i from class j is accepted at time t_i, we will refer to its *remaining processing time* at time $t \geq t_i$ as $x_i(t)$, which is initially equal to q_j for $t = t_i$, progressively decreases as the production facility dedicates processing capacity to job i, and reaches 0 upon its completion;
- As in the scheduling literature, the slack time s_j is the longest possible idle time before starting to work on a given job and still satisfy its due-date. That is, the due-date d_i for an accepted job i from class j arriving at time t_i is $d_i \equiv t_i + s_j + q_j$. Once a job i from class j is accepted, we define its remaining slack or *laxity* at time $t \geq t_i$ as $\ell_i(t) \equiv d_i - x_i(t) - t$. Note that $\ell_i(.)$ is a non-increasing function with initial value $\ell_i(t_i) = s_j$, and that a job i is completed by its due-date if and only if its laxity $\ell_i(.)$ remains non-negative on $[t_i, d_i]$;
- The profit r_j is the monetary value associated by the firm with the acceptance of a job from class j, and would typically be obtained as revenue minus cost of goods sold and/or other direct costs. The *profit rate* π_j of a job is defined as the profit obtained per unit of capacity required, that is $\pi_j = r_j/q_j$;
- We assume that the arrivals of jobs from each class j follow a Poisson process with rate

λ_j, and that the arrivals of jobs from different classes are independent. The *load* ρ of the facility can then be defined as the average total amount of potential processing time requirement arriving to the facility per unit of time, that is $\rho \equiv \sum_{j=1}^{J} \lambda_j q_j$; note that this definition relates to incoming as opposed to accepted jobs, so that the load may be larger than 1 in some market environments. Finally, we denote by λ the total arrival rate across all job classes, i.e. $\lambda \equiv \sum_{j=1}^{J} \lambda_j$.

In our model, the production facility processing each accepted job is a single machine with no setup times or setup costs. Its scheduling may at no cost be preemptive, and interrupted jobs may be resumed at no penalty. Due-dates constraints are *hard*, that is an incoming job may only be accepted if there exists a *feasible* production schedule, i.e. one satisfying the due-date of this job and also that of all other jobs already accepted but not yet completed. In addition, the production facility is constrained to always follow such a schedule. Finally, we will refer to any incoming job that may be accepted under these conditions as being *admissible*; Figure 2 contains a pictorial representation of our model.

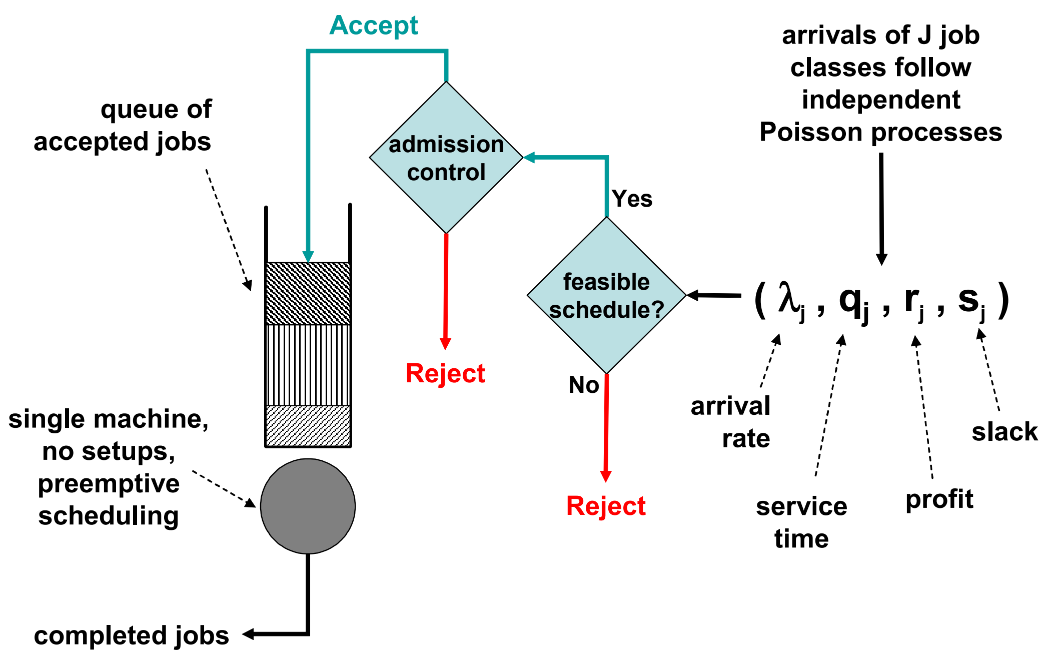

Figure 2: Graphical model representation

2.2. Dynamic Programming Formulation. In this section we present a continuous-time problem formulation in §2.2.1, then develop in §2.2.2 an alternative state representation which allows us to develop an equivalent discrete-time formulation.

2.2.1. Continuous-Time Formulation Let $n(t)$ be the number of jobs accepted but not yet completed at time t, and assume that those jobs are indexed by $i \in \{1, .., n(t)\}$. Because the job arrival process is assumed to be memoryless, the state of the system at any time t in between successive job arrivals is entirely characterized by a vector $\mathbf{M}(t) = \{(x_1(t), \ell_1(t)), ..., (x_{n(t)}(t), \ell_{n(t)}(t))\}$ describing the remaining processing time and laxity of all accepted but not yet completed jobs. In the following we will refer to vector $\mathbf{M}(t)$ as the *machine state*.

The associated scheduling or production control can be represented as a non-negative vector $\mathbf{S}(t) = \{S_1(t), ..., S_{n(t)}(t)\}$ satisfying $\sum_{i=1}^{n(t)} S_i(t) \leq 1$, where $S_i(t)$ describes the fraction of total processing capacity dedicated to job i at time t – note that the notion of assigning fractional processing capacity is consistent with our assumption that the production schedule may be preemptive-resume. More generally, a feasible production schedule associated with state $\mathbf{M}(t)$ may be formally defined as a set of $n(t)$ non-negative right-continuous functions $\mathbf{S}(.) = \{S_1(.), ..., S_{n(t)}(.)\}$ on $[t, +\infty)$ such that $\sum_{i=1}^{n(t)} S_i(\tau) \leq 1$ for $\tau \geq t$ and $\int_t^{d_i} S_i(\tau) d\tau = x_i(t)$ for all $i \in \{1, .., n(t)\}$; we will denote the set of all such feasible production schedules for $\mathbf{M}(t)$ as $\mathbb{S}[\mathbf{M}(t)]$.

Considering now a time t when a job arrives, the *system state* immediately before the acceptance/rejection decision is made can be characterized as the cartesian product $\mathbf{X}(t) = (\mathbf{M}(t), j(t))$, where $j(t) \in \{1, ..., J\}$ is the index of the class to which the incoming job belongs. The associated admission control can be represented as a binary $a(t) \in \mathbb{A}[\mathbf{X}(t)]$, with $\mathbb{A}[\mathbf{X}(t)] = \{0, 1\}$ if there exists a feasible production schedule for $\{\mathbf{M}(t), (q_{j(t)}, s_{j(t)})\}$ and $\mathbb{A}[\mathbf{X}(t)] = \{0\}$ otherwise, where $a(t) = 1$ denotes the decicion to accept a job arriving at time t. Regardless of whether an arrival occurs at time t, we define by extension the system state as $\mathbf{X}(t) = (\mathbf{M}(t), j(t))$, the set of feasible controls as $\mathbb{F}[t] = \mathbb{S}[\mathbf{M}(t)] \times \mathbb{A}[\mathbf{X}(t)]$ and the control as $\mathbf{U}(t) = (\mathbf{S}(t), a(t))$. In these extended definitions, we assume by convention that $j(t) = 0$ and $\mathbb{A}[\mathbf{X}(t)] = \{0\}$ if no arrival occurs at time t. In the following, we will overload the notation $\mathbf{U} = (\mathbf{S}, a)$, possibly referring to either the *value* of the admission and scheduling controls at time t, or to an admission control and scheduling *policy* $\mathbf{U}(\mathbf{X}(t), t) = (\mathbf{S}(\mathbf{X}(t), t), a(\mathbf{X}(t), t))$ which could be a function of system state and, in the case of a non-stationary policy, time.

More generally, we will omit any dependence on time and state when no ambiguity arises.

2.2.2. Workload Functions and Discrete-Time Formulation We begin this part by defining two concepts essential to our analysis.

Definition 1 *Consider a machine state* $\mathbf{M}$ *and feasible schedule* $\mathbf{S} \in \mathbb{S}[\mathbf{M}]$*. The cumulative workload function* $W_{\mathbf{M}}^{\mathbf{S}}(.)$ *associated with schedule* $\mathbf{S}$ *is defined for all* $\tau \geq 0$ *as*

$$W_{\mathbf{M}}^{\mathbf{S}}(\tau) \equiv \sum_{i=1}^{n} \int_{0}^{\tau} S_i(v)dv. \tag{1}$$

Intuitively, the cumulative workload function represents the total cumulative work (measured in time units) achieved by a schedule until a specified time. Note that from the definition of a feasible schedule necessarily $W_{\mathbf{M}}^{\mathbf{S}}(\tau) \leq \tau$.

Definition 2 *Consider a machine state* $\mathbf{M}$ *and assume* $\mathbb{S}[\mathbf{M}] \neq \varnothing$*. The minimum workload function* $W_{\mathbf{M}}(.)$ *associated with state* $\mathbf{M}$ *is defined for all* $\tau \geq 0$ *as*

$$W_{\mathbf{M}}(\tau) \equiv \inf_{\mathbf{S} \in \mathbb{S}[\mathbf{M}]} W_{\mathbf{M}}^{\mathbf{S}}(\tau). \tag{2}$$

When no ambiguity arises, we will simply refer to the minimum workload function (MWF) as W. Intuitively, this function represents the smallest amount of time that any schedule feasible for the current set of jobs can have worked by a given date. The reader can find an example in Figure 3, which represents the MWF corresponding to a given machine state with two jobs $\{(x_1, \ell_1), (x_2, \ell_2)\}$.

The fact that MWF functions are non-decreasing follow from definitions (1) and (2). In addition, the graph shown in Figure 3 suggests that MWF functions should in general exhibit strong continuity and differentiability properties. This is confirmed by the following proposition, which will turn out to be useful later in §3.2:

Proposition 1 *Let* $\mathbf{M} = \{(x_1, \ell_1), ..., (x_n, \ell_n)\}$ *be a feasible machine state. Then function* $W_{\mathbf{M}}$ *is continuous, piecewise linear with at most* $2n$ *break points, and the slope of its linear pieces may only be equal to either* 0 *or* 1*.*

There are at least three reasons why the MWF is of particularly interest in our setting; The first one is that it allows to characterize whether an incoming job is admissible, as shown by the following theorem:

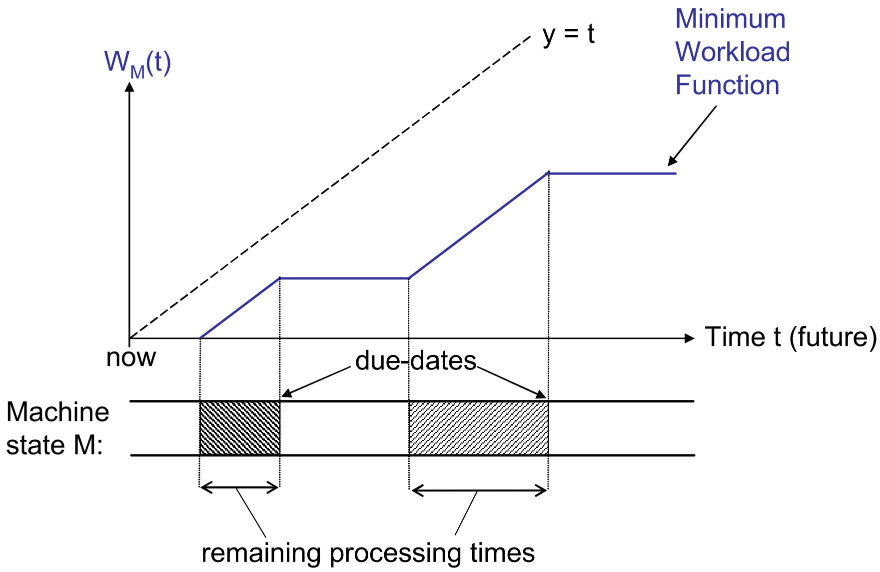

Figure 3: Example of a minimum workload function

Theorem 1 *A job i from class j arriving at time t is admissible given an existing machine state $\mathbf{M}$ (i.e. $\mathbb{S}[(\mathbf{M} \times (q_j, s_j)] \neq \varnothing$) if and only if*

$$W_{\mathbf{M}}(s_j + q_j) \leq s_j. \tag{3}$$

The intuition behind Theorem 1 is straightforward: when a new incoming job i from class j is accepted, the maximum amount of time that can be spent not working on job i by its due-date d_i is now at most equal to its slack s_j; condition (3) expresses that this slack time must be sufficient to perform the minimum amount of work required by that date on all the other jobs already accepted.

The second reason for studying the MWF is that its properties are key to a proof that always following a non-idling earliest due-date schedule ($\mathbf{ED}^*$) bears no loss of generality or cost to optimality in the setting considered, as stated in our second theorem:

Theorem 2 *Let $\mathbf{U} = (\mathbf{S}, a)$ be any feasible scheduling and admission control policy (i.e. all accepted jobs are completed by their due-dates), and let $\mathbf{U}' = (\mathbf{ED}^*, a')$ be the policy obtained by:*

- *Accepting an incoming job at t (i.e. $a'(t) = 1$) if and only if policy $\mathbf{U}$ would have accepted that job (i.e. $a(t) = 1$) when confronted with the exact same arrival process realization to date;*

- *Scheduling accepted jobs according to a non-idling earliest due-date schedule* $\mathbf{ED}^*$.

Then $\mathbf{U}'$ *is also a feasible policy, and for every arrival process realization the profit stream obtained with* $\mathbf{U}'$ *is the same as that obtained with* $\mathbf{U}$.

Theorem 2 is essential to our study, because it effectively justifies that we assume from now on a production control policy based on $\mathbf{ED}^*$ (at no cost to optimality), and solely focus our efforts on admission control. The continuous-time stochastic optimal control problem described initially thus reduces to a discrete-time dynamic program, where decisions (admission or rejection) need only be made upon job arrivals. In between job arrivals, the dynamics of the MWF are entirely characterized by the following proposition (where $[\tau]^+ \equiv \max(\tau, 0)$):

Proposition 2 *Let* $\mathbf{M}$ *be a machine state, and* $\mathbf{M}[\mathbf{ED}^*, \tau]$ *the machine state obtained applying the earliest due-date schedule* $\mathbf{ED}^*$ *on* $\mathbf{M}$ *for* τ *units of time. The graph of* $W_{\mathbf{M}[\mathbf{ED}^*,\tau]}$ *is the positive part of the graph of* $W_{\mathbf{M}}$ *translated by* $(-\tau, -\tau)$*: for all* $\tau \geq 0$ *and* $t \geq 0$,

$$W_{\mathbf{M}[\mathbf{ED}^*,\tau]}(t) = [W_{\mathbf{M}}(t+\tau) - \tau]^+. \tag{4}$$

Furthermore, the following proposition shows that under scheduling rule $\mathbf{ED}^*$ the number of jobs in any feasible machine state is bounded. Consequently, another important implication of Theorem 2 is that the state $\mathbf{X}(t) = (\mathbf{M}(t), j(t))$ has a finite dimension:

Proposition 3 *Let* $\mathbf{M} = \{(x_1, \ell_1), ..., (x_n, \ell_n)\}$ *be a feasible machine state. If* $\mathbf{M}$ *was generated by accepting jobs from* $\mathbb{J}$ *and following the earliest due-date schedule* $\mathbf{ED}^*$ *then*

$$n \leq \left\lfloor \frac{\max_{j\in\mathbb{J}} s_j}{\min_{j\in\mathbb{J}} q_j} \right\rfloor + 2.$$

Lastly, it turns out that the MWF is a sufficient state representation for our dynamic program; a precise statement follows:

Theorem 3 *Consider two initial states* $\mathbf{X}(t) = (\mathbf{M}(t), j(t))$ *and* $\mathbf{X}'(t) = (\mathbf{M}'(t), j(t))$ *such that* $W_{\mathbf{M}} = W_{\mathbf{M}'}$. *Let* a *be a feasible admission control policy, and let* a' *be the admission policy obtained by accepting an incoming job if and only if* a *does. Then for every arrival process realization beyond time* t, a' *is a feasible admission control policy, and the profit stream obtained with* a' *starting from* $\mathbf{X}'(t)$ *is the same as that obtained with* a *starting from* $\mathbf{X}(t)$.

Intuitively, Theorem 3 shows that any difference between two initial states with the same MWF bears no impact on the profit stream that can be generated with an optimal policy thereafter. Consequently, a system manager with only access to a "reduced" state information $\mathbf{Y}(t) = (W_{\mathbf{M}(t)}, j(t))$ may still make admission control decisions that are just as good as a competing decision maker with access to the full state information $\mathbf{X}(t) = (\mathbf{M}(t), j(t))$. Note that Theorem 3 amounts in fact to the statement that $\mathbf{Y}(t)$ satisfies the formal definition of a sufficient DP state statistics (Bertsekas 1995, Chapter 5), although we do not formally introduce this concept here in order to simplify exposition.

In the following, we denote $W[\tau]$ the MWF function obtained when the earliest due-date schedule $\mathbf{ED}^*$ has been applied for τ units of time on a set of jobs represented by function W. That is, the relationship between $W[\tau]$ and W is obtained by substituting $W_{\mathbf{M}[\mathbf{ED}^*,\tau]}$ with $W[\tau]$ and $W_{\mathbf{M}}$ with W in (4). Also, $W \cup j$ will denote the MWF function obtained by adding a new job from class j to the machine state represented by function W – notice from our proof of Proposition 1 that the function $W \cup j$ may be constructed from job j and function W alone (as opposed to the underlying machine state $\mathbf{M}$ such that $W = W_{\mathbf{M}}$). We are now ready to state the discrete-time formulation of our problem that will be considered in §3:

State: $\mathbf{X}_k = (W_k, j_k) \in \mathbb{X}$, where $k \in \{1, 2, ...\}$ and $j_k \in \{1, ..., J\}$ is the class to which the k-th arriving job belongs, and W_k is the MWF function corresponding to the jobs accepted in the past but not yet completed when that k-th job arrives (but before any related admission/rejection decision is made). Note that while the state space $\mathbb{X}$ is obviously not countable, Propositions 1 and 3 show that its dimension is finite and bounded by $2J\left(\lfloor \max_{j\in\mathbb{J}} s_j / \min_{j\in\mathbb{J}} q_j \rfloor + 2\right)$.

Control: $a_k \in \mathbb{A}[\mathbf{X}_k]$, with $\mathbb{A}[\mathbf{X}_k] = \{0, 1\}$ if $W_k(s_{j_k} + q_{j_k}) \le s_{j_k}$ and $\mathbb{A}[\mathbf{X}_k] = \{0\}$ otherwise, where $a_k = 1$ (resp. $a_k = 0$) denotes the decicion to accept (resp. reject) the incoming job. A feasible control policy $\mathbf{a}$ can thus be represented as $\mathbf{a} = (a_k(.))_{k\ge1}$, where a_k is a mapping from $\mathbb{X}$ to $\{0, 1\}$ such that $a_k(\mathbf{X}) \in \mathbb{A}[\mathbf{X}]$ for all $\mathbf{X} \in \mathbb{X}$.

State Dynamics: $\mathbf{X}_0 = (0, 0)$ and

$$\mathbf{X}_{k+1} = \begin{cases} ((W_k \cup j_k)[\tau], \omega) \text{ if } a_k = 1 \\ (W_k[\tau], \omega) \text{ if } a_k = 0 \end{cases} \quad \text{for } k \ge 1, \tag{5}$$

where τ is a random variable following an exponential distribution with mean λ^{-1}, and ω is a discrete random variable such that $P(\omega = j) = \lambda_j/\lambda$. From elementary properties of

Poisson processes, τ and ω can be assumed to be independent of each other, and of their own realizations at previous stages. Note that the state dynamics (5) define a probability measure[5]

$$P(\mathcal{B}|\mathbf{X}_k, a_k) = P(\mathbf{X}_{k+1} \in \mathcal{B}|\mathbf{X}_k, a_k) \tag{6}$$

associated with a σ-algebra on $\mathbb{X}$, a notation we will use from now on.

Objective: Maximize

$$C(\mathbf{a}) \equiv \liminf_{n\to+\infty} \frac{E[\sum_{k=1}^{n} a_k(\mathbf{X}_k) r_{j_k}]}{n}, \tag{7}$$

over all feasible control policies $\mathbf{a}$, which represents the long-run expected average profit per incoming job. Note that from renewal theory, the expected long-run average profit per unit of time can be obtained by multiplying $C(\mathbf{a})$ by λ, the average job arrival rate in the long run.

3. Optimization Analysis

In this section we first describe in §3.1 a characterization and some properties of the optimal policy for the problem just stated, then use that characterization in §3.2 to construct heuristic policies.

3.1. Optimal Policy Properties. We start this section with a theorem establishing in our setting the validity of the Bellman equation and the existence of an optimal stationary policy. As can be seen in the appendix, its proof essentially consists of showing that the required assumptions for Theorem 2 in Ritt and Sennott (1992) are satisfied, and invoke their result.

Theorem 4 *There exists a constant $C^* \geq 0$ and a bounded real-valued function $h(.)$ defined on $\mathbb{X}$ such that for all $\mathbf{X} = (W, j) \in \mathbb{X}$*

$$C^* + h(\mathbf{X}) = \max_{a \in \mathbb{A}[\mathbf{X}]} \left(a r_j + \int h(\mathbf{Y}) P(d\mathbf{Y}|\mathbf{X}, a) \right). \tag{8}$$

If for each $\mathbf{X} \in \mathbb{X}$, $a(\mathbf{X})$ is chosen to be the smallest action realizing the maximum on the right of (8) then the resulting stationary admission policy $\mathbf{a}^* = (a(\cdot))_k$ is optimal with expected average profit C^*, i.e.

$$C(\mathbf{a}) \leq C(\mathbf{a}^*) = C^*$$

for any feasible admission policy $\mathbf{a}$.

[5] The reader is referred to Ritt and Sennott (1992) for a mathematically rigorous presentation of the relevant DP framework.

In accordance with accepted DP terminology, in the following we will refer to a function $h(.)$ associated with an optimal stationary policy as described in Theorem 4 as a *differential value function*. Intuitively, $h(\mathbf{X})$ represents the expected difference between total profits obtained by the optimal policy when it starts from state $\mathbf{X}$, and those obtained when it starts in steady-state (where the profit obtained at each stage is C^*); $h(\mathbf{Y}) - h(\mathbf{X})$ thus represents the expected difference in total profits when the optimal policy starts from state $\mathbf{Y}$ and those obtained when it starts from state $\mathbf{X}$. Unsurprisingly, starting with a smaller queue and a more attractive incoming job entails greater future profits on average (because more future incoming jobs may be accepted), which is stated more precisely by the following proposition:

Proposition 4 *Let $\mathbf{X} = (W, j)$ and $\mathbf{X}' = (W', j')$ be two states in $\mathbb{X}$ such that $W \leq W'$, $q_j \leq q_{j'}$, $s_j \geq s_{j'}$ and $r_j \geq r_{j'}$. Then for any differential value function $h(.)$ satisfying the Bellman equation (8),*

$$h(\mathbf{X}) \geq h(\mathbf{X}'). \tag{9}$$

Under the hypotheses of Proposition 4 for jobs j and j', a direct consequence of (9) and the Bellman equation (8) is the intuitive fact that for any W

$$a(W, j) \geq a(W, j'), \tag{10}$$

where $\mathbf{a} = (a(.))_k$ is the optimal stationary policy described in Theorem 4. In words, whenever the optimal policy accepts a given job with some amount of capacity available, it would also accept a more attractive job (with more profit, smaller capacity requirements and/or a looser due-date constraint) under the same conditions. A geometric interpretation of (10) is that the optimal admission region conceptually represented in Figure 1 and defined as

$$\mathcal{A}(W) \equiv \{(q_j, s_j, r_j)_{j \in \mathbb{J}} : a(W, j) = 1\}$$

is a *cone*: For any $(j, j') \in \mathbb{J}$ such that $q_j \leq q_{j'}$, $s_j \geq s_{j'}$ and $r_j \geq r_{j'}$ then

$$(q_{j'}, s_{j'}, r_{j'}) \in \mathcal{A}(W) \Rightarrow (q_j, s_j, r_j) \in \mathcal{A}(W).$$

From both theoretical and practical standpoints, it seems relevant to determine the sub-optimality gap of the *myopic policy* consisting of accepting any feasible job. More generally, it also seems important to investigate the performance of the best *static policy*, that is a policy accepting any job from a given subset $\mathcal{S} \subset \mathbb{J}$ whenever it is feasible, or equivalently for which:

$$\mathcal{A}(W) = \{(q_j, s_j, r_j)_{j \in \mathcal{S}} : W(s_j + q_j) \leq s_j\}. \tag{11}$$

While we later address these questions through numerical experiments in §4, the following proposition establishes that the myopic policy is optimal when the load ρ is sufficiently low:

Proposition 5 *The myopic policy is optimal in any market environment* $(\lambda_j, q_j, s_j, r_j)_{j \in \mathbb{J}}$ *such that*

$$\lambda \leq \frac{1}{\bar{d}} \ln\left(\frac{\underline{r}}{-\bar{r} + \sqrt{\bar{r}^2 + 2\bar{r}\underline{r}}}\right),$$

where as before $\lambda \equiv \sum_{j \in \mathbb{J}} \lambda_j$ *and* $\bar{d} \equiv \max_{j \in \mathbb{J}}(s_j + q_j)$, $\bar{r} \equiv \max_{j \in \mathbb{J}} r_j$, $\underline{r} \equiv \min_{j \in \mathbb{J}} r_j$.

Finally, we provide an upper bound on the optimal average profit per stage C^* which only depends on the primary problem data $(\lambda_j, q_j, s_j, r_j)_{j \in \mathbb{J}}$:

Proposition 6 *The optimal average profit per stage* C^* *is bounded from above as follows:*

$$\begin{array}{rcl} C^* \leq \bar{C}^f & \equiv & \max\limits_{(a_1,\ldots,a_J)} \frac{1}{\lambda} \sum_{j \in \mathbb{J}} a_j \lambda_j r_j \\ & s.t.: & \sum_{j \in \mathbb{J}} a_j \lambda_j q_j \leq 1 \\ & & 0 \leq a_j \leq 1 \end{array} \tag{12}$$

In the following, we will refer to $\bar{C}^f$ as the *fluid upper bound*, because the formulation defining $\bar{C}^f$ in (12) would be the relevant problem to solve if the discrete and stochastic job arrivals in our market environment were replaced instead with continuous and deterministic arrivals having the same average incoming work and profit rates; the decision variables a_j in (12) represent the long-run fraction of jobs from class j admitted, while the objective expresses the corresponding long-run profit rate, and the inequality constraint is analogous to the capacity constraint in the original problem. Indeed, we find later in our numerical experiments of §4 that the bound $\bar{C}^f$ is relatively close to C^* in environments with small job *granularity*, where for a given load ρ all jobs j have a small ratio q_j/s_j. Unfortunately, the bound $\bar{C}^f$ seems to be much larger than the optimal profit C^* in many other environments, and therefore seems hardly useful in assessing the suboptimality of policies such as the ones defined by (11).

More generally, we have so far failed to characterize the optimal admission policy much beyond the results just presented. While disappointing in a way, this is hardly surprising since to date most other DP problems for which an exact solution has been presented in the literature have in essence a substantially lower state space dimension than ours. This motivates shifting our goal to developing instead heuristic policies, the focus of the next section.

3.2. Heuristic Policies. When in state $\mathbf{X}$, the optimal stationary policy $\mathbf{a}$ constructed from the Bellman equation (8):

$$\begin{cases} \text{accepts job } j \text{ if } r_j \geq \int h(\mathbf{Y})P(d\mathbf{Y}|\mathbf{X}, a=0) - \int h(\mathbf{Y})P(d\mathbf{Y}|\mathbf{X}, a=1); \\ \text{rejects job } j \text{ otherwise.} \end{cases} \tag{13}$$

Note that the (positive) r.h.s of the acceptance condition in (13) can be interpreted as the *opportunity cost* of accepting job j when in state $\mathbf{X}$, that is the expected future profits that would be specifically forgone by the immediate admission of job j. The two heuristic policies we are to present now are based on the same idea, namely to first develop an approximation for this opportunity cost, then consider the policy obtained by applying logic (13) with the approximate opportunity cost replacing the exact one; the policy presented in §3.2.1 relies on a fluid approximation, while the one presented in §3.2.2 relies on simulation and a solution to the finite horizon offline problem.

3.2.1. Fluid Policy. For any minimum workload function W and positive real number $T \geq \bar{d}$, define the *transient fluid problem* with optimal objective $F_T(W)$ as:

$$\begin{array}{rcll} F_T(W) & \equiv & \max \int_0^T \sum_{j=1}^J a_j(u)\lambda_j r_j du & \\ & \text{s.t.:} & \dot{z}_j(t) = a_j(t)\lambda_j q_j - v_j(t) & (BE) \\ & & \sum_{j=0}^J v_j(t) \leq 1 & (CP) \\ & & z_j(t) \leq \int_t^{t+s_j} v_j(u)du & (DA) \\ & & W(t) \leq \int_0^t v_0(u)du & (DE) \\ & & 0 \leq a_j(t) \leq 1 & (FR) \\ & & v_j(t) \geq 0 & (NG) \\ & & z_j(0) = 0 \text{ and } z_j(t) \geq 0, & (MO) \end{array} \tag{14}$$

where the control variables $(a_j(.), z_j(.))_{j\in\{1,\ldots,J\}}$ and $(v_j(.))_{j\in\{0,\ldots,J\}}$ are continuous and right-differentiable functions on $[0, T+\bar{s}]$, all constraints above are required to hold for all values of t in that interval, and $\dot{z}_j$ represents the right-derivative of z_j. Intuitively, the quantity $F_T(W)$ represents the maximum future profits that one could collect over the next T time units, taking into account existing delivery commitments (captured by W), and assuming that future job arrivals will be fluid (i.e. continuous and deterministic) instead of discrete and stochastic. More specifically, in formulation (14) $a_j(t)$ represents the fraction of incoming work from class j accepted at time t, $v_j(t)$ the fraction of machine capacity dedicated to working on admitted work from class j at time t, and $z_j(t)$ is the amount of work from class j outstanding at time t. While the balance equation (BE), the capacity constraint (CP), the fraction constraint (FR) and the non-negativity constraint (NG) are straightforward, other features of (14) deserve special notice:

- The objective assumes that profits are collected immediately upon admission;
- All work already committed to at time 0 and captured by the function W is assumed to belong to a special class $j = 0$. That is, $v_0(t)$ is the fraction of machine capacity dedicated to this existing commitment at time t, and in line with the definition of a MWF in (2), constraint (DE) expresses that all due-dates for this existing work should be satisfied;
- Constraint (DA) expresses that all due-dates for newly admitted work from each class $j \geq 1$ should also be satisfied. In the fluid model, each "job" corresponds to an infinitesimal quantity of work, so that the due-date of each job from class j admitted at t reduces to the relative slack $t + s_j$. Accordingly, (DA) ensures that the sojourn time (i.e. time from admission to completion) of any admitted work from class j is always smaller than this due-date;
- Constraint (MO) corresponds to our assumption of a pure make-to-order system, so that machine capacity may not be used to perform work or build inventory in anticipation of a yet unrealized demand for any job class j, which would correspond in formulation (14) to the case $z_j(t) < 0$.

Returning to our original discrete and stochastic admission control problem, we now define the *fluid policy* $\mathbf{a}^f = (a^f(.))$ in line with (13) as:

$$\begin{cases} a^f(W, j) = 1 \text{ if } r_j \geq F_T(W) - F_T(W \cup j); \\ a^f(W, j) = 0 \text{ otherwise.} \end{cases} \tag{15}$$

A primary reason why policy $\mathbf{a}^f$ is appealing is that its associated computational cost is relatively low: although not obvious from the continuous-time and variational formulation (14), computing $F_T(W)$ actually reduces to computing the optimal value of a linear program with at most $2(\lfloor \bar{s}/\underline{q} \rfloor + 3)(J+1)(2J+1)$ constraints and $2(\lfloor \bar{s}/\underline{q} \rfloor + 3)J(J+1)$ variables, as established by the following theorem.

Theorem 5 *For any minimum workload function W and number $T \geq \bar{d}$, let $U_T(W)$ be the* *union of* $\{0,T\}$ *and the set of points in* $[0,T]$ *where* $W(.)$ *is not differentiable, and for* $s \geq 0$ *define* $\mathcal{T}_s[.]$ *as the set translation operator:* $\mathcal{T}_s[A] \equiv \{x+s, x \in A\}$. *Let* $0 = \tau_0 < \tau_1 < ... < \tau_m$ *the distinct ordered points such that* $\{\tau_i : 0 \leq i \leq m\} \equiv U_T(W) \cup \left(\bigcup_{j \in \mathbb{J}} \mathcal{T}_{s_j}[U_T(W)] \right)$, *define* $\mathbb{I} \equiv \{0, ..., m\}$, *set* $w_i \equiv W(\tau_i)$ *for all* i *and consider the following linear program:*

$$\begin{aligned} LP_T(W) \quad &\equiv \quad \max_{(B_i^j)} \textstyle\sum_{j \in \mathbb{J}} \pi_j B_{T+s_j}^j \\ &s.t.: \quad w_i + \textstyle\sum_{j \in \mathbb{J}} B_i^j \leq \tau_i \text{ for all } i \in \mathbb{I} && (CA) \\ &\qquad\quad 0 \leq B_{i+1}^j - B_i^j \leq \lambda_j q_j (\tau_{i+1} - \tau_i) \text{ for all } i \in \mathbb{I} \text{ and } j \in \mathbb{J} && (AD) \\ &\qquad\quad B_i^j = 0 \text{ for all } i \in \mathbb{I} \text{ and } j \in \mathbb{J} \text{ such that } \tau_i \leq s_j. && (ST) \end{aligned} \tag{16}$$

where for notational simplicity $B_{T+s_j}^j \equiv B_i^j$ *where* i *is such that* $\tau_i = T + s_j$. *Then*

$$F_T(W) = LP_T(W). \tag{17}$$

The LP (16) is in fact a simplified, discretized and linearized version of the transient fluid problem (14), and the underlying reason why equality (17) still holds is that function W is piecewise linear (see Proposition 1). Indeed, the discretization mesh $(\tau_i)_{i \in \mathbb{I}}$ is primarily designed so that the MWF function W is linear on $[\tau_i, \tau_{i+1}]$, so that this function is entirely characterized by $(\tau_i, w_i)_{i \in \mathbb{I}}$. More specifically, each variable B_i^j in (16) represents the total amount of work admitted from class j that must be completed by time τ_i, or equivalently the total amount of work from class j admitted between time 0 and time $\tau_i - s_j$. The objective function and constraints (AD) and (ST) in (16) are straightforward in light of that interpretation, and constraint (CA) is another expression for the machine capacity constraint (a binding (CA) constraint for some $i \in \mathbb{I}$ implies that the machine never idles on $[0, \tau_i]$). Figure 4 is a graphical representation of an instance of the LP (16) with two job classes.

From a practical standpoint, Theorem 5 and definition (15) imply that the fluid policy can be implemented by keeping track of the W function over time and solving two LPs whenever a new feasible job arrives, which seems well within the reach of modern computing power. Another implication of Theorem 5 and the fact that any MWF W is constant on $[\bar{d}, +\infty)$ is that

$$\frac{\partial F_T(W)}{\partial T} = \lambda \bar{C}^f \text{ for all } T \geq \bar{d},$$

where $\bar{C}^f$ is the fluid upper bound defined in (12). Consequently, the opportunity cost approximation $F_T(W) - F_T(W \cup j)$ in (15), and thus the fluid policy, do not depend on the

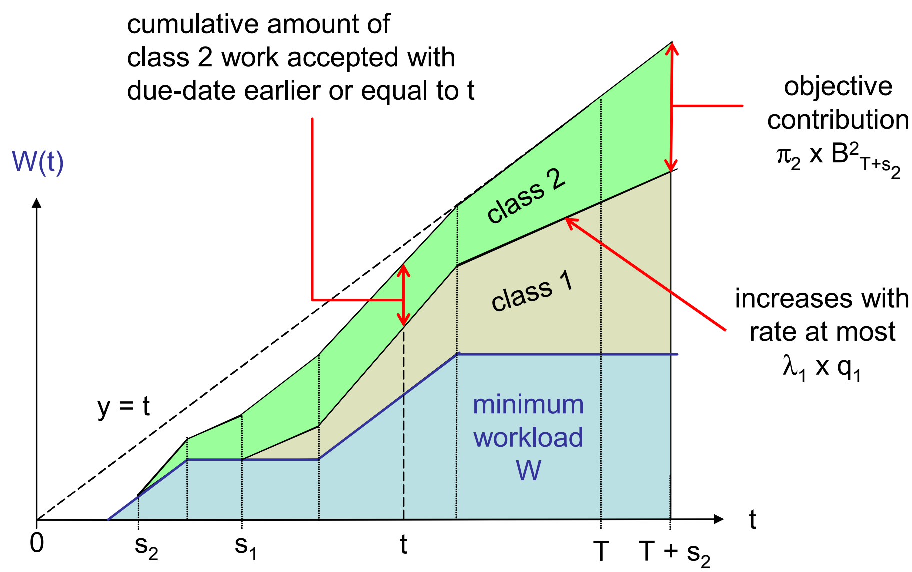

Figure 4: Graphical representation of an instance of LP (16)

choice of $T \geq \bar{d}$.

We conjecture that the fluid policy $\mathbf{a}^f$ is asymptotically optimal in the fluid limit, as stated below:

Conjecture 1 *Assume $s_j > 0$ for all $j \in \mathbb{J}$, and define a sequence of market environments $\mathbb{J}^{(n)} = (\lambda_j^{(n)}, q_j^{(n)}, s_j^{(n)}, r_j^{(n)})_{n \geq 1}$ such that $\lambda_j^{(n)} \equiv n\lambda_j$, $q_j^{(n)} \equiv q_j/n$, $s_j^{(n)} \equiv s_j$ and $r_j^{(n)} \equiv r_j/n$. If $C_{(n)}(\mathbf{a}^f)$ (resp. $C^*_{(n)}$) denote the long-run average profit per job achieved by the fluid policy $\mathbf{a}^f$ (resp. the optimal average profit per job) in environment $\mathbb{J}^{(n)}$ then*

$$\lim_{n \to +\infty} \left(C^*_{(n)} - C_{(n)}(\mathbf{a}^f) \right) / C^*_{(n)} = 0. \tag{18}$$

In the definition of the limiting market environments $\mathbb{J}^{(n)}$ above, note that the incoming work and profit arrival rates from each class j, respectively $\lambda_j^{(n)} q_j^{(n)}$ and $\lambda_j^{(n)} r_j^{(n)}$, remain identical to those in the original environment $\mathbb{J}$. Rather, the *granularity* $q_j^{(n)} \Big/ s_j^{(n)}$ of all job classes decreases as n grows large. Conjecture 1 thus draws on the observation that the main assumption on which the fluid policy relies, namely that all future job arrivals will be fluid (i.e. have infinitesimal granularity), becomes in fact closer and closer to being true as n increases. Note however that the statement of Conjecture 1 excludes environments where some jobs have zero slack, and for which the concept of granularity is thus undefined. Indeed, consider the market environment $\mathbb{J}$ with two classes $(s_1, q_1, r_1, \lambda_1) = (0, 1, 1, 1)$ and

$(s_2, q_2, r_2, \lambda_2) = (0, 1, r, 1)$ where $r < 1$. Because both classes have zero slack, incoming jobs may only be accepted when the system is empty, and a straightforward renewal process analysis shows that the optimal policy consists then of always accepting jobs from class 1, and accepting jobs from class 2 if $r \geq 1/2$. Furthermore, independently of $n \geq 1$ the condition on r for the acceptance of jobs from class 2 to be optimal in environment $\mathbb{J}^{(n)}$ (defined as in the statement of Conjecture 1) remains $r \geq 1/2$, and in that case $C^*_{(n)} = (1+r)/6n$. In contrast, for either job classes $j \in \{1, 2\}$ the approximate opportunity cost used by the fluid policy $\mathbf{a}^f$ in environment $\mathbb{J}^{(n)}$ is $F(0) - F(j) = 1/n$, so that independently of n the fluid policy consists of only accepting jobs from class 1, yielding $C_{(n)}(\mathbf{a}^f) = 1/4n$. When $r > 1/2$ the relative suboptimality of the fluid policy in environment $\mathbb{J}^{(n)}$ thus equals $(r - 1/2)/(r + 1) > 0$, so that the limit statement in (18) does not hold.

Proving (or disproving) Conjecture 1 has so far eluded us despite some intermediary progress, and we hope to resolve this issue in the future. Independently of its theoretical foundations however, the fluid policy performs relatively well in many environments and lends itself to several interesting extensions, as our numerical experiments in §4 and discussion in §5 will show.

Next, we present another admission heuristic which, if considerably more computationally intensive than the fluid policy, seems to perform at least as well in most environments, and significantly better in some.

3.2.2. Lookahead Policy. Let $\mathbf{M} = \{(x_1, \ell_1), ..., (x_n, \ell_n)\}$ be a feasible machine state and $\mathbf{\Phi} = \{(t_1, q_1, s_1, r_1), ..., (t_H, q_H, s_H, r_H)\}$ a stream of H incoming jobs now assumed to be known in advance (offline problem), where t_k denote the arrival time of job $k \in \{1, ..., H\}$. Let $\mathbb{I} \equiv \{0, ..., m\}$ and $0 = \tau_0 < \tau_1 < ... < \tau_m$ the distinct ordered points such that $\{\tau_i : i \in \mathbb{I}\} \equiv \left(\bigcup_{k=1}^{H} \{t_k, s_k + q_k\} \right) \cup \left(\bigcup_{u=1}^{n} \{\ell_u + x_u\} \right) \cup \{0\}$. Finally, for all $k \in \{1, ..., H\}$ and $u \in \{1, ..., n\}$ let α_k, δ_k and γ_u in $\mathbb{I}$ such that $\tau_{\alpha_k} = t_k$, $\tau_{\delta_k} = s_k + q_k$ and $\tau_{\gamma_u} = \ell_u + x_u$. Consider now the following mixed integer program (MIP):

$$
\begin{array}{rcll}
G(\mathbf{\Phi},\mathbf{M}) & \equiv & \max\limits_{(y_{ki},z_{ui},a_k)} \sum_{k=1}^{H} a_k r_k & \\
& \text{s.t.:} & \sum_{k=1}^{H} y_{ki} + \sum_{u=1}^{n} z_{ui} \leq \tau_i - \tau_{i-1} \text{ for all } i \in \mathbb{I}\backslash\{0\} & (CAP) \\
& & a_k q_k \leq \sum_{i=\alpha_k+1}^{\delta_k} y_{ki} \text{ for all } k \in \{1,...,H\} & (DD) \\
& & x_u \leq \sum_{i=1}^{\gamma_u} z_{ui} \text{ for all } u \in \{1,...,n\} & (DD') \\
& & y_{ki} \geq 0 \text{ and } z_{ui} \geq 0,\ a_k \in \{0,1\} & (VAR)
\end{array} \tag{19}
$$

In formulation (19), the non-negative continuous variables y_{ki} and z_{ui} represent the total amount of work performed between τ_{i-1} and τ_i on jobs k and u respectively, and each binary variable a_k represents the decision to accept of reject job k. Function $G(\mathbf{\Phi},\mathbf{M})$ is thus the maximum profit that can be made from job stream $\mathbf{\Phi}$ when the set of jobs already committed to but not yet completed is given by $\mathbf{M}$, while satisfying the machine capacity constraint (CAP) as well as the due-date contraints (DD) and (DD') of newly accepted jobs from $\mathbf{\Phi}$ and existing jobs from $\mathbf{M}$, respectively.

Note that MIP formulation (19) does not exploit the fact that scheduling (described by variables y_{ki} and z_{ui}) can be assumed with no loss of generality to follow $\mathbf{ED}^*$ in our environment (see Theorem 2). Indeed, we have found the computational requirements associated with other formulations exploiting that fact to be higher in practice than those associated with (19). Furthermore, while Lawler (1990) describes a dynamic programming algorithm solving optimization problem $G(\mathbf{\Phi},\varnothing)$ in pseudo-polynomial time, we have not attempted to extend that method to the problem with constrained admissions $G(\mathbf{\Phi},\mathbf{M})$, and use instead a general MIP branch and bound algorithm in order to solve (19). In the following, for $\epsilon \in (0,1)$ we will denote by $G_\epsilon(\mathbf{\Phi},\mathbf{M})$ the approximate objective value obtained by running that branch and bound algorithm until a solution at most ϵ suboptimal is found, i.e $\left(G(\mathbf{\Phi},\mathbf{M}) - G_\epsilon(\mathbf{\Phi},\mathbf{M})\right)/G(\mathbf{\Phi},\mathbf{M}) \leq \epsilon$.

Returning to our online problem, we now define the admission policy Lookahead(N,H,ϵ) with decisions $\mathbf{a}^{\mathrm{L}(N,H,\epsilon)} = (a^{\mathrm{L}(N,H,\epsilon)}(.))$ as:

$$
\left\{
\begin{array}{l}
a^{\mathrm{L}(N,H,\epsilon)}(\mathbf{M},j) = 1 \text{ if } r_j \geq \frac{1}{N}\sum_{k=1}^{N} G_\epsilon(\mathbf{\Phi}^{(k)},\mathbf{M}) - G_\epsilon(\mathbf{\Phi}^{(k)},\mathbf{M}\cup j); \\
a^{\mathrm{L}(N,H,\epsilon)}(\mathbf{M},j) = 0 \text{ otherwise,}
\end{array}
\right. \tag{20}
$$

where the current state is $(\mathbf{M},j)$ and $\mathbf{\Phi}^{(1)},...,\mathbf{\Phi}^{(N)}$ are N sample arrival streams of H jobs generated through simulation according to the probabilistic arrival structure defined in §2.1. In words, our lookahead policy approximates the opportunity cost for each incoming job in the online problem by a simulation-based estimation of that opportunity cost in the offline problem under a limited time horizon.

The lookahead policy just defined involves solving multiple instances of a MIP problem whenever a new feasible job arrives, and is therefore very computationally intensive. Besides, it does not involve any learning that would lessen its associated computational requirement or improve its performance over time, in contrast with other approximate dynamic programming methods (e.g. Bertsekas and Tsitsiklis 1996). It thus seems particularly important to determine values for the parameters N, H and ϵ resulting in a high policy performance $C(\mathbf{a}^{\mathrm{L}(N,H,\epsilon)})$ at a given computational cost. While solving a formal optimization model for setting (N, H, ϵ) seems out of reach, intuition suggests that the lookahead horizon H should be of the same order as $\bar{s}\lambda$. Some numerical experiments (not reported here) show indeed that the performance $C(\mathbf{a}^{\mathrm{L}})$ of Lookahead(N, H, ϵ) noticably increases with H up to a point in that vicinity, beyond which marginal performance improvements become considerably more costly from a computational standpoint. Likewise, low values of N and high values of ϵ intuitively result in the same type of error, namely reduced estimation accuracy for $E_{\mathbf{\Phi}}[G(\mathbf{\Phi}, \mathbf{M})]$; more (unreported) numerical experiments and a heuristic analysis assuming that $G_\epsilon(\mathbf{\Phi}, \mathbf{M})$ is uniformly distributed on $[(1-\epsilon)G(\mathbf{\Phi}, \mathbf{M}), G(\mathbf{\Phi}, \mathbf{M})]$ suggest that setting N close to $1/\epsilon$ is an efficient way of allocating computational resources in our problem.

As for the fluid policy, we close this subsection with a conjecture on the asymptotic optimality of Lookahead(N, H, ϵ):

Conjecture 2 Lookahead(N, H, ϵ) *is asymptotically optimal in any environment, i.e.:*

$$\lim_{\substack{N,H\to+\infty \\ \epsilon\to 0}} C(\mathbf{a}^{\mathrm{L}(N,H,\epsilon)}) = C^*.$$

We are however far less confident about Conjecture 2 than we are about Conjecture 1. Regardless of the theoretical justification that establishing this result could provide however, the experimental results we report in the next section indicate that the lookahead policy performs particularly well in all environments.

4. Numerical Experiments

The primary goal of our numerical study was to assess the relative performance of the fluid and lookahead policies described in §3.2 against that of other policies in various environments. We present our methodology in §4.1 and results in §4.2.

4.1. Methodology. The market environment $(\lambda_j, q_j, s_j, r_j)_{j\in\mathbb{J}}$ we adopted as our base case spans all combinations of $q_j \in \{1, ..., 10\}$, $s_j \in \{5, 10\}$ and $\pi_j \in \{0.7, 1\}$ (where $r_j = \pi_j q_j$), resulting in a total of $10 \times 2 \times 2 = 40$ job classes. The arrival rates λ_j are set so that $P(q_j = q) \propto 1/q^2$, $P(s_j = 5) = P(s_j = 10) = 1/2$, $P(\pi_j = 0.7) = P(\pi_j = 1) = 1/2$, the load $\rho \equiv \sum_{j=1}^{J} \lambda_j q_j$ equals 1.5 and $P(\pi_j = 1|s_j = 5) = P(\pi_j = 0.7|s_j = 10) = 3/4$ (slack and profit rate are thus negatively correlated, capturing the feature that jobs with tighter due-date requirements tend to be more profitable). It is easy to verify that the above conditions result in a linear system of equalities with a unique solution in $(\lambda_j)_{j\in\mathbb{J}}$.

In the experiments reported in §4.2, we measured how the relative performance of all policies of interest changed when some of the parameters characterizing the environment just defined were individually varied. More specifically, we explored the effects of marginal variations in the load (see §4.2.1), granularity (see §4.2.2) and profit stretch (see §4.2.3) of the market environment just defined. When investigating the relative performance of the optimal static policy in §4.2.4, we assumed instead for tractability reasons a simpler environment with only 12 classes (see that subsection for a precise definition).

We simulated the following policies in all our experiments:

Fluid: as defined in §3.2.1;

Lookahead: policy Lookahead(N, H, ϵ) defined in §3.2.2, with $(N, H, \epsilon) = (20, 20, 0.05)$;

Myopic: the static policy accepting every incoming job that is feasible;

Threshold: the static policy only accepting every incoming job with the higher profit rate (i.e. for which $\pi_j = 1$) that is feasible.

Specifically, we used a custom discrete-event simulation software written in the C programming language. When appropriate, this software relied on a dynamic, two-way data link with the solver components of CPLEX in order to generate solutions to the linear and mixed integer programs arising in these simulations. We generated 30 different simulation runs in each market environment, where each run consisted of a warm-up period of 50 job arrivals, followed by a data collection period of 200 job arrivals. We computed then the sample average and 95% confidence interval across all simulation runs for the total profit generated by each policy from those 200 jobs divided by the length (in simulation time units) of the corresponding data collection period.

In all our experiments except the granularity variation described in §4.2.2, we also report as a comparison benchmark the estimated profit rate of the **Anticipative** policy, which implements near-optimal admission/rejection decisions assuming perfect knowledge of the job arrival stream, and starts from an empty state at the beginning of the measurement period. That is, **Anticipative** is essentially the optimal offline policy in our environment (sometimes also referred to in the computer science literature as the *clairvoyant* policy), except that in order to speed up computations we only solve the associated optimization problem within 3% of optimality. Formally, if $\mathbf{\Phi} = \{(t_{51}, q_{51}, s_{51}, r_{51}), ..., (t_{250}, q_{250}, s_{250}, r_{250})\}$ denote the arrival stream of jobs from the data collection period simulated in each run, we report the average of $G_{0.03}(\mathbf{\Phi}, \varnothing)/(t_{250} - t_{50})$ across all 30 simulation runs, with function $G_\epsilon(\mathbf{\Phi}, \mathbf{M})$ as defined in §3.2.2.

Finally, we also report in §4.2.4 the simulated performance of the optimal static policy, more precisely defined in that subsection.

4.2. Results and Discussion.

4.2.1. Load Variation In this first set of experiments we investigated the effects of varying the load $\rho \equiv \sum_{j=1}^{J} \lambda_j q_j$ (average total amount of potential processing time requirement arriving to the facility per unit of time), which may serve in this setting as an indicator of the balance between supply and demand. The graph in Figure 5 represents the simulated profit rate of the first four policies mentioned in §4.1 divided by that of the **Anticipative** policy. As in all remaining graphs, error bars displayed around each data point represent the associated 95% confidence interval.

In accordance with intuition, the performance of **Myopic** decreased regularly with the load: when demand is relatively low and much capacity is available, accepting as many jobs as possible is a good policy – this is established theoretically by our Proposition 5. When demand is high relative to capacity however, the opportunity cost of accepting a job increases and one should seemingly be more selective when accepting jobs. For the exact same reasons the performance of **Threshold** increased regularly with the load: contrary to **Myopic** that

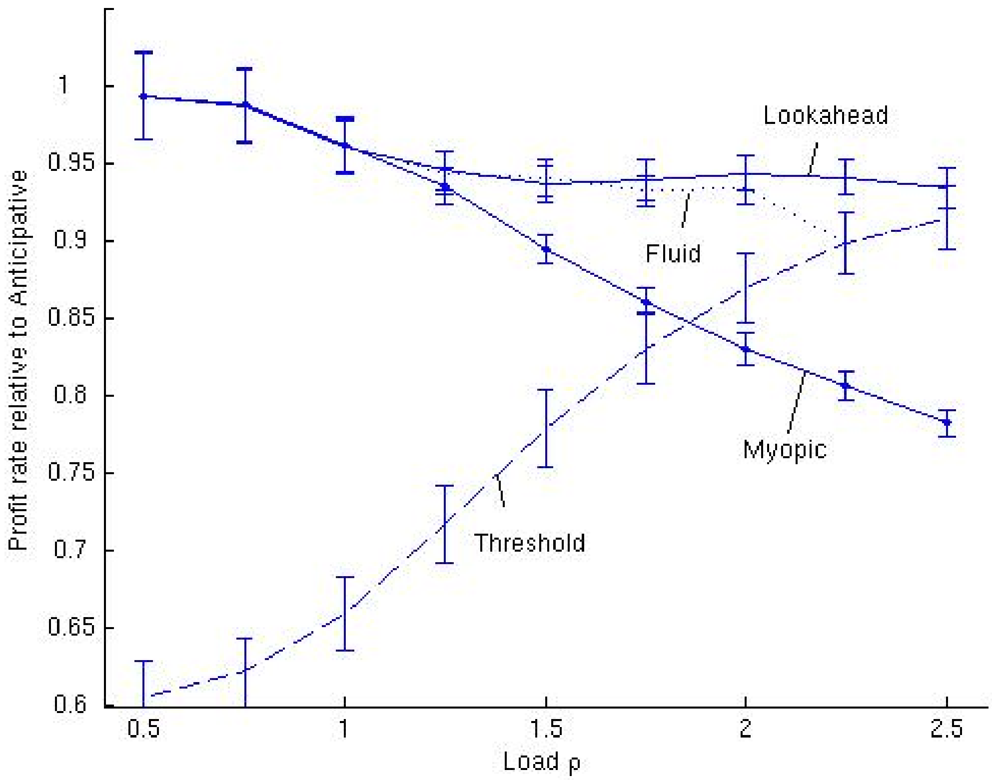

Figure 5: Load variation experiments

policy was too selective at low loads but became sensible at high loads.

The more complex **Fluid** and **Lookahead** did generally well, seemingly mimicking the behavior of **Myopic** (resp. **Threshold**) in the environments with very low (resp. high) load, that is when it is sensible to do so. For intermediate loads ranging from approximately 1.25 to 2 however, **Fluid** and **Lookahead** significantly outperformed both **Myopic** and **Threshold**, achieving more than 10% improvement over the best of those two static policies for a load slightly smaller than 2. These experiments thus suggest that the relative benefits of using sophisticated dynamic policies such as **Fluid** and **Lookahead** versus simple static policies such as **Myopic** and **Threshold** may be largest in intermediate load environments, which may arguably be more prevalent in practice. Also, the performance of **Fluid** was virtually identical to that of **Lookahead** for loads smaller than 2, a remarkable fact considering that **Lookahead** is considerably more computationally intensive than **Fluid**. For loads around 2.25 however, **Fluid** started making the same decisions as **Threshold** (this was determined by inspection of the simulation logs) and performed then noticably worse than **Lookahead**, although that performance gap seemed to reduce with higher load values of 2.5 and beyond. Unfortunately, we have not found a satisfactory explanation for this phenomenon.

Finally, the performance of **Lookahead** and **Fluid** relative to that of **Anticipative** as a function of load seemed to initially decrease until the load reached about 1.5, then slightly increase, and finally decrease again for loads larger than 2. We believe that this behavior results from the combination of two effects. The first is the value of information, which is relatively low for extreme loads (high or low), but high for intermediate loads; the second is the experimental artefact of forcing **Anticipative** to start the data collection period with an empty queue (this was imposed to ensure that its performance would constitute an upper bound). Because all the other (online) policies typically start with a non-empty queue (composed of jobs previously accepted during the warm-up period) which may restrain their ability to accept new jobs during the beginning of the data collection period, **Anticipative** derives a relative advantage from this; that advantage becomes particularly significant for high loads, where the relative impact of starting with a non-empty queue increases since the data collection period corresponds to a fixed number of jobs. In experiments (not reported here) where **Anticipative** did start with a non-empty queue (i.e. where the second effect just mentioned was eliminated), we indeed observed that its relative performance advantage over **Fluid** and **Lookahead** was unimodal, and became almost negligible for high loads.

4.2.2. Granularity Variation As part of our second set of experiments we assessed the effects of marginally varying job granularity, that is changing the ratio between processing time and slack for all incoming jobs, while keeping the load and profit rates fixed; the rationale was to explore the relative performance of our policies when the marginal impact of any individual job ranged from high (large granularity, "bulky" environment) to low (small granularity, "fluid" environment). Specifically, we simulated a serie of environments $(\lambda_j^{(n)}, q_j^{(n)}, s_j^{(n)}, r_j^{(n)})_{j\in\mathbb{J}}$ indexed by a positive real number n and derived from our base-case environment $(\lambda_j, q_j, s_j, r_j)_{j\in\mathbb{J}}$ as in the statement of Conjecture 1: $\lambda_j^{(n)} \equiv n\lambda_j$, $q_j^{(n)} \equiv q_j/n$, $s_j^{(n)} \equiv s_j$ and $r_j^{(n)} \equiv r_j/n$. Parameter n, which we refer to as the *granularity parameter*, thus provides a quantitative measure for the relative individual impact of each job (large values of n imply a fluid-like environment, while small values indicate a more bulky environment).

The graph in Figure 6 shows for various values of that parameter the performance of the first four policies mentioned in §4.1 divided by $\lambda \bar{C}^f$, where $\bar{C}^f$ is the fluid upper bound on optimal average profit per job defined in the statement of Proposition 6. Note that $\lambda \bar{C}^f$ is an upper bound on the optimal average profit per unit of time which does not depend on the granularity parameter n, so that the relative performance measures shown in Figure 6 are proportional to the absolute values of the corresponding average profit rates[6].

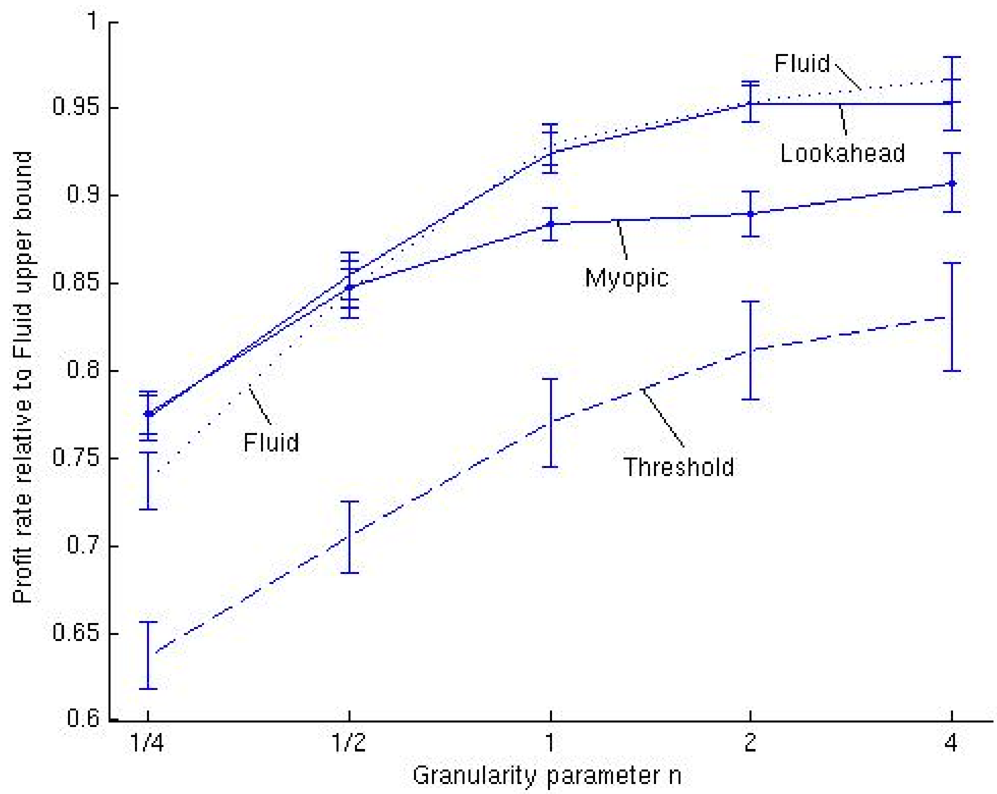

Figure 6: Job granularity variation experiments

A first observation is that the overall performance of all simulated policies improved as job granularity decreased; our interpretation is that in the fluid limit characterized by infinitesimal job granularity, the arrival process becomes deterministic, which makes for an easier market environment allowing for higher performance. On the other end, large job granularity and the resulting increased uncertainty in the arrival process of future jobs seemed to reduce

[6] We believe $\lambda \bar{C}^f$ to be a more meaningful upper bound in this setting than the profit rate of **Anticipative**, and thus used the former as a comparison benchmark for this specific set of experiments.

the opportunity cost of capacity, explaining the good performance of **Myopic** for very small values of n. **Fluid** and **Lookahead** both performed generally well in all environments, except in the high granularity limit ($n \leq 1/4$) where **Fluid** was significantly outperformed by **Myopic** and **Lookahead**. This is hardly surprising since the main assumption on which **Fluid** relies, namely that future job arrivals will be continuous and deterministic, becomes in fact increasingly invalid as n decreases. This interpretation is confirmed by the good performance of **Fluid** for small job granularity, which outperforms then all other policies including **Lookahead** for $n \geq 2$. Incidentally, we believe that the horizon length parameter $H = 20$ used for **Lookahead** may also become too small in the fluid limit, possibly accounting as well for the reduced performance of that policy relative to **Fluid** then.

Another observation about the small job granularity regime is that the performance of **Fluid** approaches the fluid upper bound as n increases; this suggests both that the fluid bound becomes increasingly tight and that **Fluid** is asymptotically optimal in the fluid limit. These results thus provide an experimental motivation for Conjecture 1, and a justification of the primary path we have been following to try and establish this result (so far unsuccessfully), namely to prove that the performance of **Fluid** converges to the fluid upper bound $\bar{C}^f$ as $n \to +\infty$.

The performance of **Myopic** was bad for small job granularity, and **Threshold** performed the worst for all values of n, although its performance seemed to improve relative to the other policies in the fluid limit. Besides the relationship between opportunity cost of capacity and arrival process uncertainty discussed earlier, our interpretation is that small job granularity induces more variation in system state upon job arrivals, so that regimes approaching the fluid limit put static policies such as **Myopic** and **Threshold** at a greater disadvantage over dynamic policies.

Finally, we point out that another related dimension characterizing our market environment is the homogeneity in the processing time requirements of incoming jobs. In some experiments (not reported here) where we simulated environments satisfying $P(q_j = q) \propto 1/q$ instead of $P(q_j = q) \propto 1/q^2$, we observed that while **Fluid** retained some statistically significant advantage over **Myopic** and **Threshold** for intermediate loads as before, **Lookahead** outperformed all three other policies more significantly than in the experiments we do report here. Our interpretation is that the increased heterogeneity in job processing time requirements introduced by the heavier tail of the distribution $1/q$ reduces the validity of the fluid arrival assumption on which **Fluid** relies, effectively amounting to larger job granularity; in these more challenging environments the more sophisticated approach for calculating opportunity cost of capacity used by **Lookahead** seemed to yield some benefits.

4.2.3. Profit Stretch Variation In a third set of experiments we investigated variations in profit stretch, defined as the ratio $\max \pi_j / \min \pi_j$ between the highest and lowest profit rate across all job classes. In our specific market environment with only two possible different profit rate values, we actually kept the higher profit rate $\max \pi_j$ at its initial value of 1 and varied the value of the smaller profit rate $\min \pi_j$ between 0.5 (high stretch environment where some jobs are twice as profitable per unit of processing time as others) and 1 (environment with no stretch where all jobs have the same profit rate). The graph representing the simulated performance of **Fluid**, **Lookahead**, **Myopic** and **Threshold** relative to that of **Anticipative** for these different values is shown in Figure 7.

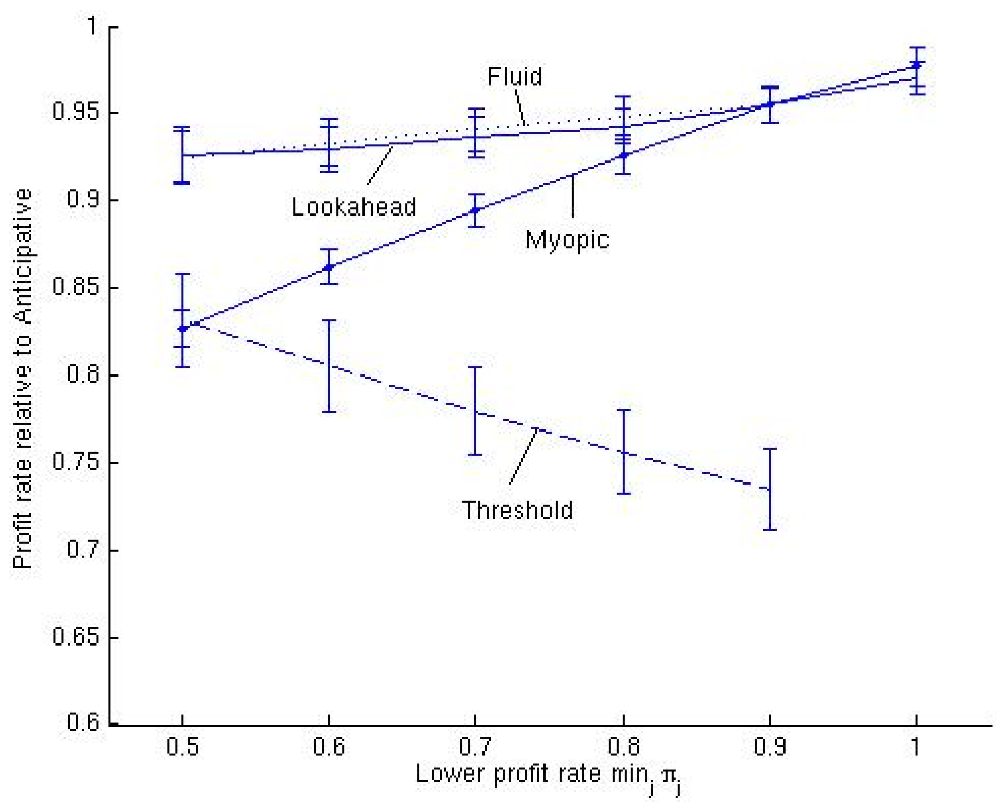

Figure 7: Profit stretch variation experiments

Fluid and **Lookahead** performed best across all stretch values, and the performances of these two policies were nearly identical overall. In the environment with no stretch ($\min \pi_j = \max \pi_j = 1$) the performances of all four policies tested were statistically indistinguishable (**Myopic** and **Threshold** are actually exactly identical in that case). In all other environments with $\min \pi_j < 1$ however, the relative performance of **Myopic** (resp. **Threshold**) decreased (resp. increased) almost linearly with stretch – the penalty from being not selective enough (resp. too selective) intuitively increases when differences in job profit rates widen (resp. diminish). While the performance of **Threshold** caught up with that of **Myopic** for the environment with the largest profit stretch we tested ($\min \pi_j = \max \pi_j/2$), the performance of either static policy remained significantly below that of our two dynamic policies then. While one may extrapolate from our results that the performance of **Threshold** may become closer to that of **Fluid** and **Lookahead** for very high profit stretch (perhaps $\min \pi_j < 0.2$), such settings are arguably not prevalent in practice. In contrast, the performance of **Myopic** became identical to that of **Fluid** and **Lookahead** in very low stretch environments ($\min \pi_j = 0.9$). For moderate profit stretch values which may be more common however, these experiments suggest that the relative benefit of using dynamic policies such as **Fluid** and **Lookahead** over simpler static policies such as **Myopic** and **Threshold** could be very significant.

4.2.4. Optimal Static Policy Finally, we designed a last set of experiments in order to assess the relative performance of the **Optimal Static** policy, defined as in §3.1 by choosing the best subset $\mathcal{S}^* \subset \mathbb{J}$ of classes in (11), that is the subset $\mathcal{S}^*$ resulting in the static policy with the highest expected average profit rate when following the non-idling earliest due-date scheduling policy **ED***; note that **Myopic** and **Threshold** correspond to the (a priori suboptimal) specific choices $\mathcal{S} = \mathbb{J}$ and $\mathcal{S} = \{j \in \mathbb{J} : \pi_j = \max \pi_j\}$, respectively.

Because we have not found to date a computationally efficient method to derive **Optimal Static** and identify set $\mathcal{S}^*$ in a general market environment, our approach for this set of experiments has been to assume a simpler market environment where it is feasible to perform an exhaustive search of all possible subsets $\mathcal{S} \subset \mathbb{J}$ through simulation. Specifically, we have investigated an environment with the same job profit rate and slack values as in our base-case scenario defined in §4.1, but with a load $\rho = 2$ and only three possible values

Policy	Profit rate
Anticipative	1.009882 ± 0.009264
Lookahead	0.950720 ± 0.013197
Fluid	0.926127 ± 0.009187
Optimal Static	0.915633 ± 0.001499
Threshold	0.895139 ± 0.026335
Myopic	0.80519 ± 0.006894

Table 1: Simulated performance of Optimal Static and other policies.

$q_j \in \{1, 2, 3\}$ for job processing time requirements, resulting in $2 \times 2 \times 3 = 12$ different job classes and $2^{12} - 1 = 4,095$ possible static policies. The average profit rate of each static policy tested was estimated with 30 simulation runs as before, but with $20,000$ job arrivals instead of 200. Although our simulation-based search does not provide any theoretical guarantee of optimality, only three other static policies among all those tested had an estimated performance within the confidence interval associated with the performance of the one we selected eventually. The policy we identified thus appears very close to being optimal among all static policies, and given our purposes we feel justified in slightly abusing terminology when still referring to it as **Optimal Static** in the following.

Interestingly, that policy consists of only accepting all high-profit jobs ($\pi_j = 1$), as well as low-profit jobs ($\pi_j = 0.7$) with *low* slack ($s_j = 5$) and processing time requirements q_j equal to either 1 or 2. Note that Proposition 4 and equation (10), which may at first seem to contradict this finding (high slack jobs may a priori seem more desirable), are only valid absent the restriction to static policies. In fact, our interpretation is that **Optimal Static** effectively achieves a desirable state-dependent behavior (accepting some low-profit jobs when the queue is empty but only high-profit jobs when it is full) by selectively including in its admission set low-profit jobs with tight due-date constraints that will thus only be feasible to admit when the system is relatively empty.

The simulated profit rate of **Optimal Static** along with that of the other policies mentioned so far in the simpler market environment just described is displayed in Table 1.

The profit rate of **Anticipative** reported in Table 1 is slightly larger than one; this results from the experimental artefact that while **Anticipative** starts the measurement period with an empty state, it frequently ends it with a full queue. Among online policies, the performance of **Optimal Static** came within 91.5% of optimality, and the relative performance superiority of **Lookahead** over **Fluid** (+2.6%) was larger than that of **Fluid** over **Optimal Static** (+1.1%). This suggests that the performance achievable with **Optimal Static** may in general come relatively close to that attained by more sophisticated dynamic policies. However, because these results are based on a very specific, simple and possibly pathological environment, they should be treated with considerably more caution than our other findings. We note that while the optimal static policy may be computationally intensive to derive, it is however particularly simple to implement in practice. These observations motivate our future research goals of further validating the results presented here and finding a computationally efficient method to derive the optimal static policy in general market environments.

5. Implementation Issues and Conclusion

In Sections §2, §3 and §4, we have investigated the dynamic admission control of jobs with hard deadlines and deterministic processing time from a finite set of classes into a single machine queue with preemptive scheduling. Using the concept of minimum workload function (MWF), we have established that earliest due-date scheduling can be assumed at no cost to optimality, and developed a discrete-time formulation for the problem of maximizing long-run expected profit. We have exploited a characterization of the optimal policy for this problem in order to derive two heuristic policies (*fluid* and *lookahead*) relying on different approximations for the opportunity cost of accepting a job. While the lookahead policy is very computationally intensive as it involves solving several mixed integer programs upon each job arrival, the fluid policy leverages the special structure of MWFs and thus only requires solving two LPs then. Numerical experiments under various load, stretch and granularity parameters suggest that they perform better than simple static policies in almost all environments. Limited experiments also suggest that the optimal static policy performs nearly as well as these two dynamic heuristics in some settings. However, while the optimal static policy is extremely simple to implement, it does seem computationally intensive to derive, at least using the method we have identifed so far (simulation-based optimization). Overall, our fluid heuristic stands out for its robust performance at a relatively low computational cost. That heuristic is also attractive because its extension in practice to the case of non-stationary demand seems straightforward: assuming that the average demand rate for each demand class j is now given by a (time-dependent) function $\lambda_j(\tau)$, one may introduce a set of points $(\tau_i)_{i\in\mathbb{I}'}$ and values $\lambda_i^j \equiv \lambda_j((\tau_i + \tau_{i+1})/2)$ such that the function $\sum_{i\in\mathbb{I}'} \lambda_i^j 1_{[t_i,t_{i+1}]}$ constitutes an acceptable piece-wise constant approximation of $\lambda_j(.)$, and substitute constraint (AD) in the LP (16) defining the fluid policy with

$$0 \leq B_{i+1}^j - B_i^j \leq \lambda_i^j q_j(\tau_{i+1} - \tau_i) \text{ for all } i \in \mathbb{I} \text{ and } j \in \mathbb{J} \qquad (AD_t) \ ,$$

where the time discretization mesh $(\tau_i)_{i\in\mathbb{I}}$ is obtained by merging the one defined in the statement of Theorem 5 with $(\tau_i)_{i\in\mathbb{I}'}$. Another appealing possible extension in practice concerns orders with multiple staggered deliveries: denote by $W \cup_{k=1}^h j_k$ the MWF corresponding to both the current set of jobs not yet completed with MWF W and a new order with h deliveries $\cup_{k=1}^h j_k$, with $j_k = (s_k, q_k)$ characterizing the requirements of the k-th delivery; the opportunity cost for that new order may be approximated for practical purposes by $F(W) - F(W \cup_{k=1}^h j_k)$, where $F(.)$ is the function defined by LP (16). While the theoretical grounding of this method would undoubtedly benefit from further research, it may thus also constitute a useful tool for the pricing of contracts (as opposed to individual transactions).

Returning to the original motivation for the admission control model investigated, namely the development of dynamic salesforce guidelines, we observe that the structure (15) of the fluid policy seems well suited to computing the portion of the admission surface defined in §1 that is relevant during an interaction with a potential customer. For example, if $j = (s, q)$ represent the processing requirements and due-date requested by a customer when the facility state is W, the quantity $F(W) - F(W \cup j)$ could suggest the lowest price that a sales agent may consent to. More generally, active quotations and counter offers represented by the projections in Figure 1 may be determined through mere line searches starting from any partial information provided by the customer about the transaction of interest; for a price-sensitive customer initially requesting an unprofitable transaction $j_0 = (r, s_0, q)$, such line search may take the form of exploring slack values $(s_k)_{k\geq 1}$defining $j_k = (s_k, q)$ in order to find s_K such that $F(W) - F(W \cup j_K) = r$. Note that the opportunity cost $F(W) - F(W \cup j_k)$ is non-increasing in s_k, and that while the first step in this line search involves solving two LPs (to compute $F(W)$ and $F(W \cup j_0)$, every subsequent step only requires solving a single one (to compute $F(W \cup j_k)$). Also, even when such terminating s_K is determined to not exist, the minimum value of $F(W) - F(W \cup j_k)$ found during the search may still suggest a price and lead-time combination constituting a sensible counter-offer.

As stated in §1, make-to-order transactions often involve in practice additional terms beyond price, lead-time and quantity such as payment delays, warranty, insurance, return policy, etc. This may clearly limit the applicability of the work presented in markets where such additional terms are non-standard or constitute a sensitive component of the corresponding transaction contracts. However, in situations where such additional terms are fairly standard and the range of their possible values limited (e.g. "Net 15", "Net 30" or "Net 45" for payment terms), one may consider defining baseline values for all those additional contract terms, use our methods to price transaction on the basis of price, lead-time and quantity under the assumption that the additional terms for all future transaction opportunities would be identical to these baseline values, and then account for any actual deviation in those terms with pre-determined multiplicative or additive relative price corrections.

It is also sometimes feared that formal sales guidelines may hamper the ability to discriminate between customers for reasons not reflected in the terms of any given transaction. In practice, some firms will indeed offer advantageous sales condition to customers perceived as "strategic", for example when it is believed that they could generate many other sales in the future, or because of the publicity and/or credibility created when dealing with a well-known "blue chip" customer. These practices are often supported by a well-thought out rationale of market segmentation, and may be very sensible from the perspective of maximizing long-run profits. We observe that our methods amount to establishing likewise a market segmentation, but one based on relative price and delivery-lead time sensitivity for individual transactions as opposed to long-run customer relationship value: we implicitly assume a spot market environment, and our theoretical model does not recognize that several different transactions may originate from the same customer. Such firms may define customer categories based on relationship value, and use our model to establish a baseline from which relative price corrections may be applied according to these categories. In situations where such discriminations between customers constitute the norm rather than the exception however, more suited models may be in order. Also, the possibility described above that our method for approximating opportunity cost may also apply to contracts (in addition to single transactions) would seem particularly worthy of further investigation in these environments.

Perhaps the strongest limitation of the results presented here stems from our representation of a production facility as a single machine with preemptive scheduling and no set-ups. While there are several examples in practice where this model does closely approximates reality (magnetic resonance imaging, photolithography masks manufacturing[7], commercial DNA sequencing), there are many other make-to-order systems for which it is clearly unrealistic. In the future, we hope to study production models involving set-ups and multiple machines, but also investigate the use of our single machine model for approximating transaction opportunity costs in more complex environments. Specifically, we envision combining our algorithms with a type of ERP software component known as "Available-To-Promise" (ATP), which essentially provides a feasibility check (or a completion date prediction) for potential transactions given complex equipment and labor capacity constraints as well as inventory availability restrictions and, when appropriate, raw material delivery schedules. Our initial thought is to use such software to not only determine the feasibility of any potential transaction, but also construct through multiple queries a capacity curve for a complex production system conceptually similar to the MWF we have introduced in this paper for a single-machine model; one possible approximation technique could then consist of treating that capacity curve with the algorithms presented in §3.2 as though it had been generated by a single-machine system, and compute the corresponding opportunity cost. This prospective approximation scheme is illustrated in Figure 8.

Finally, we hope to conduct more theoretical work allowing to resolve the conjectures stated in Sections §3.2.1 and §3.2.2, namely that our fluid and lookahead heuristics are asymptotically optimal in the fluid and infinite lookahead horizon limits, respectively. Also, we intend to research better methods for deriving the optimal static admission policy, and explore more systematically the performance gap between the optimal static policy and more sophisticated dynamic policies such as the ones presented in this paper.

Appendix

We caution the reader that the proof of Proposition 2 is presented in this appendix before that of Theorem 2; this is because the proof of Theorem 2 relies on Proposition 2, even

[7] We thank Pr. Michael J. Harrison for bringing this example to our attention.

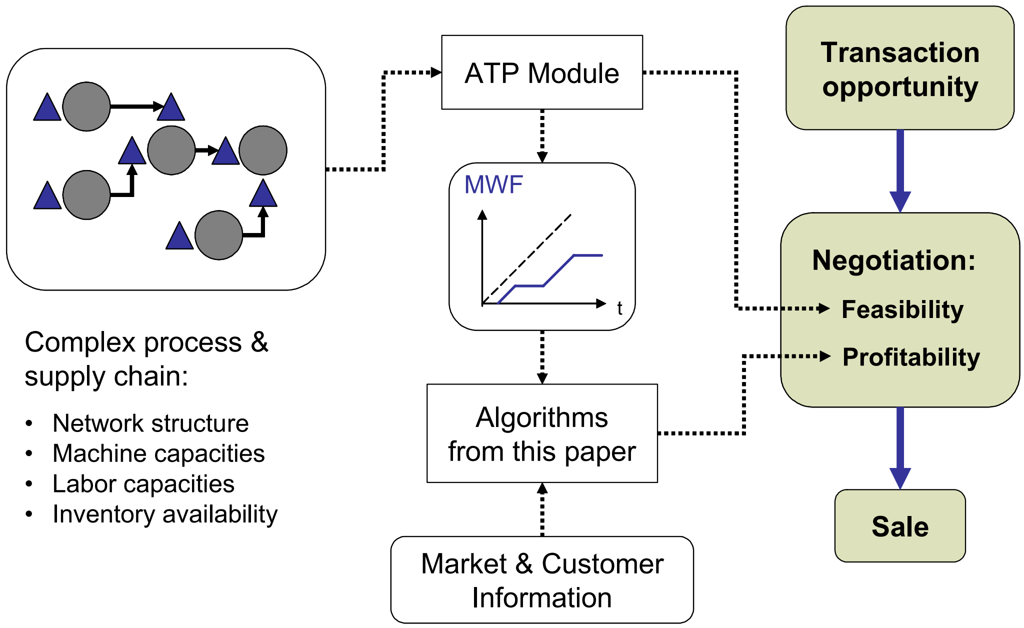

Figure 8: Prospective approximation scheme for complex production systems

though we found it more natural to reverse the presentation order of these two results in the text.

A.1. Proof of Proposition 1. Proposition 1 is actually the direct consequence of Proposition 8, a deeper result providing a characterization of MWF functions which is stated and proved later in this section of the appendix. We first need to prove some other intermediary results describing important properties of earliest due-date schedules and cumulative workload functions:

Proposition 7 *Let* $\mathbf{M} = \{(x_1, \ell_1), ..., (x_n, \ell_n)\}$ *be a feasible machine state (i.e.* $\mathbb{S}[\mathbf{M}] \neq \varnothing$*). Define the set* $\mathbb{E}[\mathbf{M}]$ *of earliest due-date schedules as the set of all schedules* $\mathbf{E}(.) = \{E_1(.), ..., E_n(.)\}$ *where* $E_1, ..., E_n$ *are non-negative, right-continuous functions on* $[0, +\infty)$ *satisfying the property:*

$$\left\{ \begin{array}{l} d_i = x_i + \ell_i < x_k + \ell_k = d_k \\ E_k(\tau) > 0 \end{array} \right. \quad \Rightarrow \int_0^\tau E_i(v)dv = x_i. \tag{21}$$

Then for any feasible schedule $\mathbf{S} \in \mathbb{S}[\mathbf{M}]$*, there exists an earliest due date schedule* $\mathbf{E} \in \mathbb{E}[\mathbf{M}]$ *such that*

$$W_{\mathbf{M}}^{\mathbf{E}} = W_{\mathbf{M}}^{\mathbf{S}}.$$

Proof. This proof relies on an adaptation to our setting of an idea used independently by Dertouzos (1974) and Horn (1974) to prove that the earliest due date schedule (also known as Jackson's rule) solves $1|pmtn; r_i|L_{\max}$. Assume $\mathbf{S} \notin \mathbb{E}[\mathbf{M}]$ and let $i, k \in \{1, ..., n\}$ and

$\tau < d_i$ such that $d_i < d_k$, $S_k(\tau) > 0$ and $\int_0^\tau S_i(v)dv < x_i$. Because S_i and S_k are right-continuous, we can assume that $P_k(\tau) \equiv \int_0^\tau S_k(v)dv > 0$ and $R_i(\tau) \equiv \int_\tau^{d_i} S_i(v)dv > 0$, since $\mathbf{S}$ is feasible therefore $\int_0^{d_i} S_i(v)dv = x_i = R_i(\tau) + \int_0^\tau S_i(v)dv$. Note that P_k is continuous and non-decreasing with $P_k(0) = 0$, while R_i is continuous and non-increasing with $R_i(0) = x_i$ and $R_i(d_i) = 0$. Therefore $P_k(d_i) \geq P_k(\tau) > 0$, so that the function $R_i - P_k$ is continuous and changes sign on $[0, d_i]$, which implies the existence of $\tau' \in (0, d_i)$ such that $R_i(\tau') = P_k(\tau')$. Define now the schedule $\mathbf{S}'$ by:

$$\begin{cases} S'_m = S_m \text{ for } m \neq i, k \\ \begin{cases} S'_i(v) = S_i(v) + S_k(v), \ \forall v \in [0, \tau') \\ S'_i(v) = 0, \ \forall v \in [\tau', d_i] \end{cases} \\ \begin{cases} S'_k(s) = 0, \ \forall v \in [0, \tau') \\ S'_k(s) = S_i(v) + S_k(v), \ \forall s \in [\tau', d_i) \\ S'_k(s) = S_k(s), \ \forall s \in [d_i, d_k] \end{cases} \end{cases} \tag{22}$$

The component functions of S' are non-negative, right-continuous, and satisfy $\sum_i^n S'_i(v) \leq 1$ for all $v \geq 0$, in addition we have by construction:

$$\begin{aligned} \int_0^{d_i} S'_i(v)dv &= \int_0^{\tau'} S'_i(v)dv + \int_{\tau'}^{d_i} S'_i(v)dv \\ &= \int_0^{\tau'} (S_i(v) + S_k(v))\, dv \\ &= \int_0^{\tau'} S_i(v)dv + P_k(\tau') \\ &= \int_0^{\tau'} S_i(v)dv + R_i(\tau') \\ &= \int_0^{d_i} S_i(v)dv \\ &= x_i, \end{aligned}$$

and

$$\begin{aligned}
\int_0^{d_k} S'_k(v)dv &= \int_0^{\tau'} S'_k(v)dv + \int_{\tau'}^{d_i} S'_k(v)dv + \int_{d_i}^{d_k} S'_k(v)dv \\
&= \int_{\tau'}^{d_i} (S_i(v) + S_k(v))\, dv + \int_{d_i}^{d_k} S_k(v)dv \\
&= R_i(\tau') + \int_{\tau'}^{d_i} S_k(v)dv + \int_{d_i}^{d_k} S_k(v)dv \\
&= P_k(\tau') + \int_{\tau'}^{d_k} S_k(v)dv \\
&= \int_0^{d_k} S_k(v)dv \\
&= x_k,
\end{aligned}$$

so that $\mathbf{S}'$ is feasible, i.e. $\mathbf{S}' \in \mathbb{S}[\mathbf{M}]$. In addition, $\int_0^{\tau'} S'_i(v)dv = x_i$ and $S'_k = 0$ on $[0, \tau']$, so that condition (21) is now satisfied for indices i and k by $\mathbf{S}'$. Observe that the same construction procedure can be applied iteratively for any other pair of indices for which $\mathbf{S}'$ may still violate (21), and that there can be at most a finite number of such pairs. Because from (22) that construction also preserves the cumulative workload function, i.e. $W_{\mathbf{M}}^{\mathbf{S}'} = W_{\mathbf{M}}^{\mathbf{S}}$, the proof is complete. ■

Corollary 1 *Let* $\mathbf{M}$ *be a machine state, and* $\mathbf{ED}^*$ *be as before the non-idling earliest due-date schedule.*

$$\mathbb{S}[\mathbf{M}] \neq \varnothing \Leftrightarrow \mathbf{ED}^* \in \mathbb{S}[\mathbf{M}]$$

Proof. The "if" part of this result is trivial. For the "only if" part, note that Dertouzos (1974) actually proves a more general result in a scheduling model with release dates. To prove this simpler version in our setting it suffices now to notice that Proposition 7 implies

$$\begin{aligned}
\mathbb{S}[\mathbf{M}] \neq \varnothing &\Rightarrow \mathbb{E}[\mathbf{M}] \cap \mathbb{S}[\mathbf{M}] \neq \varnothing \\
&\Rightarrow \mathbf{ED}^* \in \mathbb{S}[\mathbf{M}],
\end{aligned}$$

where the second implication holds because if any (possibly idling) earliest due-date schedule is feasible, then $\mathbf{ED}^*$ also is.

■

Proposition 8 *Let* $\mathbf{M} = \{(x_1, \ell_1), ..., (x_n, \ell_n)\}$ *be a feasible machine state (i.e.* $\mathbb{S}[\mathbf{M}] \neq \varnothing$*), where we assume (with no loss of generality) that the jobs are ordered by increasing due-dates, that is* $d_i = x_i + \ell_i \leq x_{i+1} + \ell_{i+1} = d_{i+1}$*. Let* $\mathbf{\Sigma}(\mathbf{M}) = \{\Sigma_1, ..., \Sigma_n\} \in \mathbb{E}[\mathbf{M}]$ *be the maximum-idling earliest due date schedule of* $\mathbf{M}$*, defined by the starting time* b_i *and ending time* e_i *of work by* $\mathbf{\Sigma}$ *on each job* i *as follows:*

$$\Sigma_i(\tau) = \mathit{1}_{[b_i, e_i]}(\tau), \text{ where } (b_i, e_i) \text{ is given by:}$$
$$\begin{cases} e_n = d_n \\ e_{i-1} = \min(d_{i-1}, e_i - x_i) \text{ for } i \in \{2, ..., n\} \\ b_i = e_i - x_i \text{ for } i \in \{1, ..., n\} \end{cases} . \tag{23}$$

Then $\mathbf{\Sigma} \in \mathbb{S}[\mathbf{M}]$ *(i.e.* $\mathbf{\Sigma}$ *is feasible), and the cumulative workload function of* $\mathbf{\Sigma}$ *achieves the minimum workload function of* $\mathbf{M}$*, that is*

$$W_{\mathbf{M}}^{\mathbf{\Sigma}(\mathbf{M})} = W_{\mathbf{M}}.$$

Proof. It follows from (23) that $\mathbf{\Sigma} \in \mathbb{S}[\mathbf{M}]$ if and only if $b_1 \geq 0$. System (23) implies $b_1 = \min(d_1 - x_1, d_2 - x_2 - x_1, ..., d_n - \sum_{i=1}^n x_i)$. Let k be the index in that expression such that $b_1 = d_k - \sum_{i=1}^k x_i$. The strict inequality $b_1 < 0$ would thus imply $\int_0^{d_k} ED_k^*(v)dv < x_k$ therefore $\mathbf{ED}^* \notin \mathbb{S}[\mathbf{M}]$, which from Corollary 1 would imply $\mathbb{S}[\mathbf{M}] = \varnothing$, a contradiction. This proves $\mathbf{\Sigma} \in \mathbb{S}[\mathbf{M}]$.

Turning to the second statement, Proposition 7 implies that for all $\tau \geq 0$,

$$\inf_{\mathbf{S} \in \mathbb{S}[\mathbf{M}]} W_{\mathbf{M}}^{\mathbf{S}}(\tau) = \inf_{\mathbf{E} \in \mathbb{E}[\mathbf{M}] \cap \mathbb{S}[\mathbf{M}]} W_{\mathbf{M}}^{\mathbf{E}}(\tau). \tag{24}$$

Let $\mathbf{E} = \{E_1, ..., E_n\} \in \mathbb{E}[\mathbf{M}] \cap \mathbb{S}[\mathbf{M}]$ be a feasible earliest due-date schedule; From (21) there exist $\{(b'_1, e'_1), ..., (b'_n, e'_n)\}$ such that $E_i = 0$ on $[0, +\infty) \backslash [b'_i, e'_i]$, $b'_i < e'_i \leq b_{i+1}$ and $e'_i - b'_i \geq x_i$ for all $i \in \{1, ..., n\}$. Besides, it follows from the construction (23) of $\mathbf{\Sigma}$ that $e'_i \leq e_i$ and $b'_i \leq b_i$ for all i. We now prove by contradiction that

$$\int_0^\tau E_i(v)dv \geq \int_0^\tau \Sigma_i(v)dv \text{ for all } \tau \geq 0. \tag{25}$$

Otherwise, let τ such that $\int_0^\tau E_i(v)dv < \int_0^\tau \Sigma_i(v)dv$; since $\mathbf{E} \in \mathbb{S}[\mathbf{M}]$ we have $\int_0^e E_i(v)dv = x_i \geq \int_0^e \Sigma_i(v)dv$ for $e \geq e'_i$, therefore $\tau < e'_i \leq e_i$. Also $\int_0^\tau E_i(v)dv \geq 0$ implies $\int_0^\tau \Sigma_i(v)dv > 0$ therefore $\tau > b_i$ and $\int_0^\tau \Sigma_i(v)dv + (e_i - \tau) = x_i$. But then

$$\begin{aligned}\int_{\tau}^{e_i'} E_i(v)dv &\leq e_i' - \tau \\ &\leq e_i - \tau \\ &= x_i - \int_0^{\tau} \Sigma_i(v)dv \\ &< x_i - \int_0^{\tau} E_i(v)dv,\end{aligned}$$

implying $\int_0^{e_i'} E_i(v)dv < x_i$, which contradicts $\mathbf{E} \in \mathbb{S}[\mathbf{M}]$ and proves (25). Summing inequalities (25) for all i yields

$$W_{\mathbf{M}}^{\mathbf{E}}(\tau) \geq W_{\mathbf{M}}^{\mathbf{\Sigma}}(\tau) \text{ for all } \tau \geq 0,$$

which combined with (24) finally proves that $W_{\mathbf{M}} = W_{\mathbf{M}}^{\mathbf{\Sigma}}$. ■

Proposition 1 now directly follows from Proposition 8, since this last result implies the existence of $\{(b_1, e_1), ..., (b_n, e_n)\}$ such that $b_i < e_i \leq b_{i+1} < e_{i+1}$ for $i \in \{1, ..., n-1\}$ and

$$W_{\mathbf{M}}(\tau) = \sum_{k=1}^{n} \int_0^{\tau} 1_{[b_k, e_k]}(\tau)d\tau$$

for all $\tau \geq 0$.

A.2. Proof of Theorem 1. The proof of Theorem 1 relies on the following chain of equivalent statements, in which the first "only if" follows from Proposition 8, and the second from the construction $S_{n+1}(\tau) = 1 - \sum_{i=1}^{n} S_i(\tau)$ on $[0, \bar{\tau}]$ where $\bar{\tau}$ is such that $\int_0^{\bar{\tau}} (1 - \sum_{i=1}^{n} S_i(\tau))\, d\tau = q_j$. All "if" statements are straightforward.

$$\inf_{\mathbf{S} \in \mathbb{S}[\mathbf{M}]} W_{\mathbf{M}}^{\mathbf{S}}(q_j + s_j) \leq s_j$$

$$\Leftrightarrow \exists \{S_1(.), ..., S_n(.)\} \text{ on } [0, +\infty) \text{ such that} \left\{ \begin{array}{l} \int_0^{x_i+\ell_i} S_i(\tau)d\tau = x_i \\ \sum_{i=1}^{n} S_i(\tau) \leq 1 \end{array} \right. \text{, and} \int_0^{q_j+s_j} \sum_{i=1}^{n} S_i(\tau)d\tau \leq s_j$$

$$\Leftrightarrow \exists \{S_1(.), ..., S_{n+1}(.)\} \text{ on } [0, +\infty) \text{ such that} \sum_{i=1}^{n+1} S_i(\tau) \leq 1 \text{ and } \left\{ \begin{array}{l} \int_0^{x_i+\ell_i} S_i(\tau)d\tau = x_i \\ \int_0^{q_j+s_j} S_{n+1}(\tau)d\tau = q_j \end{array} \right.$$

$$\Leftrightarrow \mathbb{S}[(\mathbf{M} \cup (q_j, s_j)] \neq \varnothing$$

A.3. Proof of Proposition 2. Let $\mathbf{M} = \{(x_1, \ell_1), ..., (x_n, \ell_n)\}$ and assume (with no loss of generality) that the jobs in $\mathbf{M}$ are ordered by increasing due-dates. Because schedule

$\mathbf{ED}^*$ is an earliest due-date schedule, there exists $k \in \{1, ..., n\}$ such that $\mathbf{M}[\mathbf{ED}^*, \tau] = \{(x'_k, \ell'_k), ..., (x'_n, \ell'_n)\}$ with

$$\begin{cases} x'_j + \ell'_j = x_j + \ell_j - \tau \text{ for } j \in \{k, ..., n\} \\ x'_j = x_j \text{ for } j \in \{k+1, ..., n\} \end{cases} . \tag{26}$$

In addition, because $\mathbf{ED}^*$ is non-idling necessarily

$$\tau = \sum_{j=1}^{k-1} x_i + x_k - x'_k. \tag{27}$$

Let now (b_j, e_j) (resp. (b'_j, e'_j)) be the beginning and end times of work by $\mathbf{\Sigma}(\mathbf{M})$ (resp. $\mathbf{\Sigma}(\mathbf{M}[\mathbf{ED}^*, \tau])$) on job $j \in \{k, ..., n\}$. From the recursion (23) defining $\mathbf{\Sigma}(\mathbf{M})$ and $\mathbf{\Sigma}(\mathbf{M}[\mathbf{ED}^*, \tau])$, we have

$$\begin{cases} e'_j = e_j - \tau \text{ for all } j \in \{k, ..., n\} \\ b'_j = b_j - \tau \text{ for all } j \in \{k+1, ..., n\} \\ b'_k = e_j - \tau - x'_k = b_k + x_k - x'_k - \tau \end{cases} . \tag{28}$$

From Proposition 8 we can now write

$$\begin{aligned} W_{\mathbf{M}[\mathbf{ED}^*,\tau]}(t) &= \sum_{j=k}^{n} \int_0^t 1_{[b'_j, e'_j]}(v)dv \\ &= \sum_{j=k+1}^{n} \int_0^t 1_{[b_j-\tau, e_j-\tau]}(v)dv + \int_0^t 1_{[e_j-\tau-x'_k, e_k-\tau]}(v)dv \\ &= \sum_{j=k+1}^{n} \int_\tau^{t+\tau} 1_{[b_j, e_j]}(v)dv + \int_\tau^{t+\tau} 1_{[e_j-x'_k, e_k]}(v)dv \\ &= \sum_{j=k+1}^{n} \int_0^{t+\tau} 1_{[b_j, e_j]}(v)dv + \int_0^{t+\tau} 1_{[e_j-x'_k, e_k]}(v)dv \\ &= W_{\mathbf{M}}(t+\tau) - \sum_{j=1}^{k} \int_0^{t+\tau} 1_{[b_j, e_j]}(v)dv + \int_0^{t+\tau} 1_{[e_j-x'_k, e_k]}(v)dv \\ &= W_{\mathbf{M}}(t+\tau) - \sum_{j=1}^{k-1} \int_0^{t+\tau} 1_{[b_j, e_j]}(v)dv - \int_0^{t+\tau} 1_{[b_k, b_k+x_k-x'_k]}(v)dv \end{aligned} \tag{29}$$

where the first equality follows from Proposition 8, the second from (28), the third from the change of variable $v + \tau \to v$, the fourth from the fact that the feasibility of $\mathbf{M}$ implies $b_j \geq \tau$ for $j \in \{k+1, ..., n\}$ and $e_j - x'_k \geq \tau$, the fifth from Proposition 8, and the sixth from combining the last term of the summation with the last integral. Define now the following functions

$$\begin{cases} F_{\mathbf{M}}(T) = \sum_{j=1}^{k-1} \int_0^T 1_{[b_j,e_j]}(v)dv + \int_0^T 1_{[b_k,b_k+x_k-x'_k]}(v)dv \\ G_{\mathbf{M}}(T) = \int_0^T 1_{[b_k+x_k-x'_k,e_k]}(v)dv + \sum_{j=k+1}^{n} \int_0^T 1_{[b_j,e_j]}(v)dv \end{cases};$$

it follows from Proposition 8 that

$$W_{\mathbf{M}}(t+\tau) = F_{\mathbf{M}}(t+\tau) + G_{\mathbf{M}}(t+\tau), \tag{30}$$

and from (29) that

$$W_{\mathbf{M}[\mathbf{ED}^*,\tau]}(t) = W_{\mathbf{M}}(t+\tau) - F_{\mathbf{M}}(t+\tau). \tag{31}$$

In addition, the construction of $\mathbf{\Sigma}(\mathbf{M})$ implies $e_j \leq b_{j+1}$ for $j \in \{1, ..., n\}$, so the construction of $F_{\mathbf{M}}$ and $G_{\mathbf{M}}$, the fact that $e_j - b_j = x_j$ and equation (27) imply in turn:

$$\begin{cases} 0 \leq F_{\mathbf{M}}(t+\tau) \leq \tau \\ G_{\mathbf{M}}(t+\tau) > 0 \Rightarrow t+\tau > b_k + x_k - x'_k \Rightarrow F_{\mathbf{M}}(t+\tau) = \tau \end{cases};$$

it follows therefore from (30) that $F_{\mathbf{M}}(t+\tau) = \min(W_{\mathbf{M}}(t+\tau), \tau)$, and substituting into (31) finally yields

$$\begin{aligned} W_{\mathbf{M}[\mathbf{ED}^*,\tau]}(t) &= W_{\mathbf{M}}(t+\tau) - \min(W_{\mathbf{M}}(t+\tau), \tau) \\ &= [W_{\mathbf{M}}(t+\tau) - \tau]^+. \end{aligned}$$

A.4. Proof of Proposition 3. Define $\bar{s} \equiv \max_{j \in \mathbb{J}} s_j$, $\underline{q} \equiv \min_{j \in \mathbb{J}} q_j$, and for each original job admission stream Γ consider a modified job admission stream Γ' where the job admission times remain the same but where each accepted job (q_j, s_j) is now replaced with $(\underline{q}, \bar{s})$. Note that if Γ is feasible then Γ' also is, because each accepted job in Γ' requires less work and has a looser due-date constraint than its counterpart in Γ. Consequently, the maximum number of jobs accepted but not yet completed over all feasible policies and arrival streams in environment $\mathbb{J} = (q_j, s_j)_{j \in \{1,\ldots,J\}}$, if it exists, is smaller than that for environment $\mathbb{J}' = (\underline{q}, \bar{s})_{j \in \{1,\ldots,J\}}$. But observe that in $\mathbb{J}'$ schedule ED^* works on admitted jobs in a first-in-first-out manner, because all jobs have the same relative due-date $\bar{s} + \underline{q}$. As a result, in any feasible machine state obtained under $\mathbb{J}'$ and following ED^*, all jobs except one have a quantity of work remaining equal to $\underline{q}$. From Theorem 1, the number of jobs m in a machine state where it is feasible to accept a new incoming job thus satisfies

$$
\begin{aligned}
(m-1)\underline{q} &\leq \bar{s} \\
\Rightarrow m &\leq \frac{\bar{s}}{\underline{q}}+1 \\
\Rightarrow m &\leq \left\lfloor \frac{\bar{s}}{\underline{q}} \right\rfloor +1,
\end{aligned} \tag{32}
$$

where the last inequality follows from the fact that m is integer. Combining (32) with the last remark of the previous paragraph, we have therefore proven that the maximum number n of jobs accepted but not yet completed in environment $\mathbb{J}$ does exist, and is smaller than $\lfloor \bar{s}/\underline{q} \rfloor + 2$.

A.5. Proof of Theorem 2. We first need some intermediary results. In the following, we use the notation $W_{\mathbf{M}_1} \leq W_{\mathbf{M}_2}$ when $\mathbf{M}_1$ and $\mathbf{M}_2$ are two machine states such that $W_{\mathbf{M}_1}(t) \leq W_{\mathbf{M}_2}(t)$ for all t. Also, if $\mathbf{M}$ is a machine state and $\mathbf{S} \in \mathbb{S}[\mathbf{M}]$ is a feasible schedule of $\mathbf{M}$, we note $\mathbf{M}[\mathbf{S}, \tau]$ the machine state obtained after schedule $\mathbf{S}$ works on the jobs in $\mathbf{M}$ for τ units of time. Finally, if $\mathbf{j} = (q_j, s_j)$ is an admissible job for $\mathbf{M} = \{(x_1, \ell_1), ..., (x_n, \ell_n)\}$, we note $\mathbf{M} \cup \mathbf{j} = \{(x_1, \ell_1), ..., (x_n, \ell_n), (q_j, s_j)\}$ the machine state obtained when $\mathbf{j}$ is accepted.

Lemma 1 *Let* $\mathbf{M} = \{(x_1, \ell_1), ..., (x_n, \ell_n)\}$ *be a machine state and* $\mathbf{j} = (q_j, s_j)$ *an admissible job. Then* $W_{\mathbf{M}} \leq W_{\mathbf{M}\cup\mathbf{j}}$.

Proof. From Proposition 8 $W_{\mathbf{M}} = W_{\mathbf{M}}^{\mathbf{\Sigma}(\mathbf{M})}$ and $W_{\mathbf{M}\cup\mathbf{j}} = W_{\mathbf{M}\cup\mathbf{j}}^{\mathbf{\Sigma}(\mathbf{M}\cup\mathbf{j})}$, where $\mathbf{\Sigma}(\mathbf{M})$ and $\mathbf{\Sigma}(\mathbf{M}\cup\mathbf{j})$ are the maximum-idling earliest due-date schedules for $\mathbf{M}$ and $\mathbf{M} \cup \mathbf{j}$. Let:

- (b_k, e_k) the beginning and end times for work on job $k \in \{1, ..., n\}$ in the construction of $\mathbf{\Sigma}(\mathbf{M})$; and
- (b'_k, e'_k) the beginning and end times for work on job $k \in \{1, ..., n\} \cup \{j\}$ in the construction of $\mathbf{\Sigma}(\mathbf{M} \cup \mathbf{j})$.

Assume (with no loss of generality) that all jobs $k \in \{1, ..., n\}$ with the same deadline as j (i.e. $x_k + \ell_k = q_j + s_j$) are given lower index numbers than for job j in the construction (23) of $\mathbf{\Sigma}(\mathbf{M} \cup \mathbf{j})$, but the relative order of jobs $k \in \{1, ..., n\}$ in the construction of $\mathbf{\Sigma}(\mathbf{M} \cup \mathbf{j})$ is the same as that in the construction of $\mathbf{\Sigma}(\mathbf{M})$ otherwise. It follows then from the recursion

(23) that $b'_k \leq b_k$ and $e'_k \leq e_k$ for all $k \in \{1, ..., n\}$, implying for all $\tau \geq 0$:

$$\begin{aligned} W_{\mathbf{M}}(\tau) &= W_{\mathbf{M}}^{\mathbf{\Sigma}(\mathbf{M})}(\tau) \\ &= \sum_{k=1}^{n} \int_0^{\tau} \mathbf{\Sigma}_k(\mathbf{M})(\tau) d\tau \\ &= \sum_{k=1}^{n} \int_0^{\tau} 1_{[b_k, e_k]}(\tau) d\tau \\ &\leq \sum_{k=1}^{n} \int_0^{\tau} 1_{[b'_k, e'_k]}(\tau) d\tau \\ &\leq \sum_{k=1}^{n} \int_0^{\tau} 1_{[b'_k, e'_k]}(\tau) d\tau + \int_0^{\tau} 1_{[b'_j, e'_j]}(\tau) d\tau \\ &= W_{\mathbf{M} \cup \mathbf{j}}(\tau), \end{aligned}$$

which concludes the proof. ■

Lemma 2 *Let $\mathbf{M_1}$ and $\mathbf{M_2}$ be two feasible machine states such that $W_{\mathbf{M_1}} \leq W_{\mathbf{M_2}}$, and $\mathbf{j}$ $= (q_j, s_j)$ an admissible job in both states. Then $W_{\mathbf{M_1} \cup \mathbf{j}} \leq W_{\mathbf{M_2} \cup \mathbf{j}}$.*

Proof. In the construction (23), note that schedule $\mathbf{\Sigma}$ will work continuously on a set of jobs $\{k, k+1, ..., k+m\} \subset \{1, ..., n\}$ with contiguous indices when $e_i = b_{i+1}$ for all $i \in \{k, k+1, ..., k+m-1\}$. We define the periods $[b_k, e_{k+m}]$ corresponding to the largest possible such sets of indices as the *busy periods* of schedule $\mathbf{\Sigma}$.

Let now (B_j, E_j) be the beginning and end times of the busy period containing job j in schedule $\mathbf{\Sigma}(\mathbf{M_1} \cup \mathbf{j})$; note that because $\mathbf{j}$ is assumed to be admissible necessarily $s_j + q_j \leq E_j$. Also, it follows from the construction of $\mathbf{\Sigma}$ in (23) that the beginning and end times of jobs in schedule $\mathbf{\Sigma}(\mathbf{M_1} \cup \mathbf{j})$ that do not belong to $[B_j, E_j]$ are the same as the beginning and end times for those same jobs in schedule $\mathbf{\Sigma}(\mathbf{M_1})$. From Proposition 8 this implies:

$$W_{\mathbf{M_1} \cup \mathbf{j}}(t) = W_{\mathbf{M_1}}(t) \text{ for all } t \leq B_j. \tag{33}$$

In addition, the beginning and end times of jobs k in $\mathbf{M_1}$ (resp. $\mathbf{M_1}$) with a due-date $x_k + \ell_k$ strictly larger than $d_j = s_j + q_j$ in schedule $\mathbf{\Sigma}(\mathbf{M_1} \cup \mathbf{j})$ (resp. $\mathbf{\Sigma}(\mathbf{M_2} \cup \mathbf{j})$) are also the same as the beginning and end times for those same jobs in schedule $\mathbf{\Sigma}(\mathbf{M_1})$ (resp. $\mathbf{\Sigma}(\mathbf{M_2})$). From

Proposition 8 this implies:

$$\begin{cases} W_{\mathbf{M_1}\cup\mathbf{j}}(t) = W_{\mathbf{M_1}}(t) + q_j \\ W_{\mathbf{M_2}\cup\mathbf{j}}(t) = W_{\mathbf{M_2}}(t) + q_j \end{cases} \text{for all } t \geq d_j \tag{34}$$

We now show that the inequality $W_{\mathbf{M_1}\cup\mathbf{j}}(t) \leq W_{\mathbf{M_2}\cup\mathbf{j}}(t)$ holds in all three intervals $[0, B_j]$, $[B_j, d_j]$ and $[d_j, +\infty)$:

- If $t \leq B_j$,

$$\begin{aligned} W_{\mathbf{M_1}\cup\mathbf{j}}(t) &= W_{\mathbf{M_1}}(t) \\ &\leq W_{\mathbf{M_2}}(t) \\ &\leq W_{\mathbf{M_2}\cup\mathbf{j}}(t), \end{aligned}$$

  where the first equalities follows from (33), the first inequality from the Lemma statement assumption and the second inequality from Lemma 1.

- If $t \geq d_j$, it follows from the Lemma assumption and (34) that

$$\begin{aligned} & W_{\mathbf{M_1}}(t) \leq W_{\mathbf{M_2}}(t) \\ \Leftrightarrow\; & W_{\mathbf{M_1}}(t) + q_j \leq W_{\mathbf{M_2}}(t) + q_j \\ \Leftrightarrow\; & W_{\mathbf{M_1}\cup\mathbf{j}}(t) \leq W_{\mathbf{M_2}\cup\mathbf{j}}(t). \end{aligned} \tag{35}$$

- Let $t \in [B_j, d_j]$, since $d_j \leq E_j$ schedule $\mathbf{\Sigma}(\mathbf{M_1} \cup \mathbf{j})$ is non-idling on $[t, d_j]$, therefore $W_{\mathbf{M_1}\cup\mathbf{j}}(t) > W_{\mathbf{M_2}\cup\mathbf{j}}(t)$ would imply

$$\begin{aligned} W_{\mathbf{M_1}\cup\mathbf{j}}(d_j) &= W_{\mathbf{M_1}\cup\mathbf{j}}(t) + d_j - t \\ &> W_{\mathbf{M_2}\cup\mathbf{j}}(t) + d_j - t \\ &\geq W_{\mathbf{M_2}\cup\mathbf{j}}(d_j), \end{aligned}$$

  where the second inequality follows from $\sum_k \mathbf{\Sigma}_k(\mathbf{M_2} \cup \mathbf{j})(\tau) \leq 1$ for all τ. But this would contradict (35) which is valid in particular for $t = d_j$. ■

Lemma 3 *Let* $\mathbf{M}$ *be a feasible machine state and* $\mathbf{j} = (q_j, s_j)$, $\mathbf{j'} = (q_{j'}, s_{j'})$ *two admissible jobs such that* $q_j \leq q_{j'}$ *and* $s_j \geq s_{j'}$. *Then* $W_{\mathbf{M}\cup\mathbf{j}} \leq W_{\mathbf{M}\cup\mathbf{j'}}$.

Proof. It follows from the construction (23) that if b_i (resp. b'_i) is the beginning time of work by $\Sigma(\mathbf{M} \cup \mathbf{j})$ (resp. $\Sigma(\mathbf{M} \cup \mathbf{j'})$) on any job i, then $b'_i \leq b_i$. ■

Proposition 9 *Let* $\mathbf{M}$ *be a machine state,* $\mathbf{S} \in \mathbb{S}[\mathbf{M}]$ *an arbitrary feasible schedule for* $\mathbf{M}$ *and* $\mathbf{ED}^*$ *the non-idling earliest due-date schedule. Then the MWF for the state obtained following* $\mathbf{ED}^*$ *for any time* τ *is no larger than that obtained following* $\mathbf{S}$*: For all* $\tau \geq 0$*,*

$$W_{\mathbf{M}[\mathbf{ED}^*,\tau]} \leq W_{\mathbf{M}[\mathbf{S},\tau]}.$$

Proof. Let $\mathbf{M} = \{(x_1, \ell_1), ..., (x_n, \ell_n)\}$ with $x_1 + \ell_1 = d_i \leq d_{i+1} = x_{i+1} + \ell_{i+1}$, and let e_i be the ending time of work by schedule $\Sigma(\mathbf{M}[\mathbf{ED}^*, \tau])$ on job i as defined in recursion (23). By contradiction, assume that there exists $t > 0$ such that

$$W_{\mathbf{M}[\mathbf{S},\tau]}(t) < W_{\mathbf{M}[\mathbf{ED}^*,\tau]}(t). \tag{36}$$

If t belongs to a busy period of schedule $\Sigma(\mathbf{M}[\mathbf{ED}^*, \tau])$, then the contradiction hypothesis (36) still holds at the end $\bar{t}$ of that busy period, since from Proposition (1) $W_{\mathbf{M}[\mathbf{ED}^*,\tau]} - W_{\mathbf{M}[\mathbf{S},\tau]}$ is non-decreasing on $[t, \bar{t}]$. Likewise, if t belongs to an idle period of $\Sigma(\mathbf{M}[\mathbf{ED}^*, \tau])$ (36) still holds at the end $\underline{t}$ of the last busy period before t (which exists since (36) implies $W_{\mathbf{M}[\mathbf{ED}^*,\tau]}(t) > 0$), because $W_{\mathbf{M}[\mathbf{ED}^*,\tau]} - W_{\mathbf{M}[\mathbf{S},\tau]}$ is non-increasing on $[\underline{t}, t]$. We may therefore assume that t is the end of a busy period for schedule $\Sigma(\mathbf{M}[\mathbf{ED}^*, \tau]$, which from (23) implies that there exists a job $j \in \{1, ..., n\}$ with a due-date at time τ which is equal to t, or equivalently such that $t = d_j - \tau$. Because schedule $\mathbf{ED}^*$ is non-idling, this and Proposition (8) imply

$$W_{\mathbf{M}[\mathbf{ED}^*,\tau]}(t) = \sum_{i=1}^{j} q_i - \tau. \tag{37}$$

Let now τ_i be the amount of time spent by schedule $\mathbf{S}$ on job $i \in \{1, ..., n\}$ up until time τ. The capacity constraint implies $\sum_{i=1}^{n} \tau_i \leq \tau$ so a fortiori $\sum_{i=1}^{j} \tau_i \leq \tau$, and necessarily

$$\begin{aligned} W_{\mathbf{M}[\mathbf{S},\tau]}(t) &< W_{\mathbf{M}[\mathbf{ED}^*,\tau]}(t) \\ &= \sum_{i=1}^{j} q_i - \tau \\ &\leq \sum_{i=1}^{j} (q_i - \tau_i) \end{aligned} \tag{38}$$

where the first inequality is the contradiction hypothesis, the equality is given by (37) and the last inequality follows from the preceding remark. But notice that the r.h.s. $\sum_{i=1}^{j} (q_i - \tau_i)$ of (38) is the total amount of work remaining on jobs in $\mathbf{M}[\mathbf{S}, \tau]$ with a due-date smaller than t, which from the definition (2) implies

$$W_{\mathbf{M}[\mathbf{S},\tau]}(t) \geq \sum_{i=1}^{j} (q_i - \tau_i)\,,$$

a contradiction; the proof of Proposition 9 is complete. ■

The next result follows directly from Proposition 2:

Corollary 2 *Let $\mathbf{M_1}$ and $\mathbf{M_2}$ be two feasible machine states such that $W_{\mathbf{M}_1} \leq W_{\mathbf{M}_2}$. Then the MWF for the state obtained applying $\mathbf{ED}^*$ on $\mathbf{M_1}$ for any time τ remains no larger than that obtained applying $\mathbf{ED}^*$ on $\mathbf{M_2}$: For all $\tau \geq 0$,*

$$W_{\mathbf{M}_1[\mathbf{ED}^*,\tau]} \leq W_{\mathbf{M}_2[\mathbf{ED}^*,\tau]}.$$

We are now ready to state the proof of Theorem 2, which consists of showing by induction on the number k of jobs arrived to date that, when confronted with the same sample path of job arrivals, the MWF obtained when following policy $\mathbf{U}$ is always larger than or equal to the MWF obtained when following policy $\mathbf{U}'$. From Theorem 1, this implies in particular that policy $\mathbf{U}'$ is feasible, and (by construction) that it yields the exact same revenue stream as that obtained with $\mathbf{U}$.

Specifically, let $t_k > 0$ be the time of the k-th job arrival (we define $t_0 = 0$ for convenience) and $\mathbf{M}(t)$ (resp. $\mathbf{M}'(t)$) be the machine state obtained at time $t \geq 0$ when following policies $\mathbf{U}$ (resp. $\mathbf{U}'$). By convention, we will assume that the admission of a job $\mathbf{j}_k$ at time t_k impacts the machine state immediately after, but not at, time t_k. That is,

$$\begin{cases} W_{\mathbf{M}(t_k^-)} \equiv \lim_{\substack{t\to t_k \\ t<t_k}} W_{\mathbf{M}(t)} = W_{\mathbf{M}(t_k)} \\ W_{\mathbf{M}(t_k^+)} \equiv \lim_{\substack{t\to t_k \\ t>t_k}} W_{\mathbf{M}(t)} = W_{\mathbf{M}(t_k)\cup\mathbf{j}_k} \end{cases},$$

and if job $\mathbf{j}_k$ is rejected at time t_k then $W_{\mathbf{M}(t_k^-)} = W_{\mathbf{M}(t_k^+)} = W_{\mathbf{M}(t_k)}$.

Our induction hypothesis is that

$$\text{for all } k \geq 1 \text{ and } t \in [t_{k-1}, t_k],\ W_{\mathbf{M}'(t)} \leq W_{\mathbf{M}(t)}. \tag{39}$$

Since the first arrival is assumed to occur at $t_1 > 0$, the base of the induction follows from the fact that $W_{\mathbf{M}'(t)} = W_{\mathbf{M}(t)} = \mathbf{0}$ for $t \in [0, t_1]$. Assume now that (39) holds up until a given k; from Theorem 1 it is feasible for policy $\mathbf{U}$ to accept job $\mathbf{j}_k = (q_k, s_k)$ arriving at time t_k if and only if $W_{\mathbf{M}(t_k)}(s_k + q_k) \leq s_k$. But the induction hypothesis implies

$W_{\mathbf{M}'(t_k)}(s_k+q_k) \leq W_{\mathbf{M}(t_k)}(s_k+q_k)$, therefore if it is feasible for $\mathbf{U}$ to accept $\mathbf{j}_k$ then it is also feasible for $\mathbf{U}'$ to do so. In addition, Lemma 2 implies

$$W_{\mathbf{M}'(t_k)\cup\mathbf{j}_k} \leq W_{\mathbf{M}(t_k)\cup\mathbf{j}_k},$$

so that we have proven

$$W_{\mathbf{M}'(t_k^+)} \leq W_{\mathbf{M}(t_k^+)} \tag{40}$$

when job $\mathbf{j}_k$ is accepted, and inequality (40) follows directly from (39) when $\mathbf{j}_k$ is rejected. For all $t \in (t_k, t_{k+1}]$ we can thus write

$$\begin{aligned} W_{\mathbf{M}'(t)} &= W_{\mathbf{M}'(t_k^+)[\mathbf{ED}^*, t-t_k]} \\ &\leq W_{\mathbf{M}(t_k^+)[\mathbf{ED}^*, t-t_k]} \\ &\leq W_{\mathbf{M}(t_k^+)[\mathbf{S}, t-t_k]} \\ &= W_{\mathbf{M}(t)}, \end{aligned}$$

where the first equality follows from the definition of $\mathbf{U}'$, the first inequality from (40) and Corollary 2, the second inequality from Proposition 9, and the last equality from the definition of $\mathbf{U}$; the proof of Theorem 2 is complete.

A.6. Proof of Theorem 3. Theorem 3 is proven by specializing the assumptions of Theorem 2 to the particular case $\mathbf{U} = (\mathbf{ED}^*, a)$ and $\mathbf{U}' = (\mathbf{ED}^*, a')$, and adapting its proof to show that $W_{\mathbf{M}'(\tau)} = W_{\mathbf{M}(\tau)}$ for all $\tau \geq t$. Specifically, define now t_k as the time of the k-th arrival after time t, and let $t_0 = t$. The arguments for the base of the induction and for the general induction step are now identical: Applying Lemma 2 twice for the two inequalities $W_{\mathbf{M}'(t_k)} \leq W_{\mathbf{M}(t_k)}$ and $W_{\mathbf{M}(t_k)} \leq W_{\mathbf{M}'(t_k)}$ shows that $W_{\mathbf{M}'(t_k)} = W_{\mathbf{M}(t_k)}$ implies $W_{\mathbf{M}'(t_k)\cup\mathbf{j}_k} = W_{\mathbf{M}(t_k)\cup\mathbf{j}_k}$ or

$$W_{\mathbf{M}'(t_k^+)} = W_{\mathbf{M}(t_k^+)}. \tag{41}$$

Furthermore for $u \geq 0$ and $\tau \in (t_k, t_{k+1}]$,

$$\begin{aligned} W_{\mathbf{M}'(\tau)}(u) &= W_{\mathbf{M}'(t_k^+)[\mathbf{ED}^*,\tau-t_k]}(u) \\ &= [W_{\mathbf{M}'(t_k^+)}(u+\tau-t_k)-\tau+t_k]^+ \\ &= [W_{\mathbf{M}(t_k^+)}(u+\tau-t_k)-\tau+t_k]^+ \\ &= W_{\mathbf{M}(t_k^+)[\mathbf{ED}^*,\tau-t_k]}(u) \\ &= W_{\mathbf{M}(\tau)}(u), \end{aligned}$$

where the first and last equalities are tautological, the second and fourth equalities follow from Proposition 2, and the third equality from (41) or, for the base of the induction, from the Theorem statement hypothesis that $W_{\mathbf{M}'(t)} = W_{\mathbf{M}(t)}$; the proof of Theorem 3 is now complete.

A.7. Proof of Theorem 4. Our proof consists of showing that Assumptions 3 and 4 of Theorem 2 in Ritt and Sennott (1992) hold (its Assumptions 1 and 2 are obviously satisfied in the present setting), and show that a slightly stronger version of their result applies to our specific model. For any feasible control policy $\mathbf{a} = (a_k(\cdot))_{k\geq 0}$, state $\mathbf{X} \in \mathbb{X}$ and discount factor $\alpha \in (0,1)$ define

$$V_\alpha(\mathbf{a},\mathbf{X}) \equiv \lim_{N\to+\infty} E\left[\sum_{k=0}^{N} \alpha^k a_k(\mathbf{X}_k) r_{j_k} \middle| \mathbf{X}_0 = \mathbf{X}\right] \tag{42}$$

and

$$V_\alpha(\mathbf{X}) \equiv \sup_{\mathbf{a}} V_\alpha(\mathbf{a},\mathbf{X}), \tag{43}$$

where the sup in (43) is taken over all feasible control policies. Note that these two functions are well-defined since the term under the expectation operator in (42) is bounded from above by $\bar{r}/(1-\alpha)$ with $\bar{r} \equiv \max_{j\in\mathbb{J}} r_j$. Besides, Theorem 1 in Ritt and Sennott (1992) imply the existence of an α-discount optimal policy, that is a policy achieving the supremum in (43) for all $\mathbf{X}$. That the abovementioned Assumptions 3 and 4 are satisfied now directly follows from the inequality

$$|V_\alpha(\mathbf{X}') - V_\alpha(\mathbf{X})| \leq \frac{\bar{r}}{\exp(-2\lambda\bar{d})} \text{ for all } (\mathbf{X}',\mathbf{X}) \in \mathbb{X}, \tag{44}$$

where $\bar{d} \equiv \max_{j\in\mathbb{J}} (s_j + q_j)$. We now prove (44): consider an α-discount optimal control policy $\mathbf{a}^* = (a_k^*(\cdot))_{k\geq 0}$ applied to two systems Γ and Γ' starting from initial states $\mathbf{X}_0 = \mathbf{X}$

and $\mathbf{X}'_0 = \mathbf{X}'$ respectively, but faced with the same job arrival stream thereafter (i.e. for $k \geq 1$). Define random variable $T \geq 0$ as the smallest index such that $W'_T = W_T = 0$; note that the time between two consecutive job arrivals is exponential with mean λ^{-1} thus

$$P(T = k | T \geq k) \geq e^{-\lambda \bar{d}} \text{ for } k \geq 0,$$

therefore

$$P(T \geq k) \leq (1 - e^{-\lambda \bar{d}})^k. \tag{45}$$

Also, the respective states $\mathbf{X}'_k$ and $\mathbf{X}_k$ and discounted profit streams in systems Γ and Γ' are identical for $k \geq T$. Thus

$$\begin{aligned}
|V_\alpha(\mathbf{X}') - V_\alpha(\mathbf{X})| &= |V_\alpha(\mathbf{a}^*, \mathbf{X}') - V_\alpha(\mathbf{a}^*, \mathbf{X})| \\
&= |a_0^*(\mathbf{X}') r_{j'_o} - a_0^*(\mathbf{X}) r_{j_o} \\
&+ \sum_{k \geq 2} P(T = k) E \left[\sum_{t=1}^{k-1} \alpha^t r_{j_t} (a_t^*(\mathbf{X}'_t) - a_t^*(\mathbf{X}_t)) \middle| T = k \right] | \\
&\leq \bar{r} + \sum_{k \geq 2} P(T = k)(k-1)\bar{r} \\
&\leq \bar{r} \sum_{k \geq 0} P(T \geq k)(k+1) \\
&\leq \bar{r} \sum_{k \geq 0} (1 - e^{-\lambda \bar{d}})^k (k+1) \\
&\leq \frac{\bar{r}}{\exp(-2\lambda \bar{d})},
\end{aligned}$$

where the penultimate inequality follows from (45); this completes the proof of (44) and achieves to show that Theorem 2 in Ritt and Sennott applies. In particular, they show the existence of a sequence $(\alpha_n)_{n \geq 1}$ in $(0, 1)$ with $\lim_{n \to +\infty} \alpha_n = 1$ such that for any state $\mathbf{Y}$, the constructs

$$\begin{cases} h(\mathbf{X}) \equiv \limsup_{n \to +\infty} (V_{\alpha_n}(\mathbf{X}) - V_{\alpha_n}(\mathbf{Y})) \\ C^* \equiv \lim_{n \to +\infty} (1 - \alpha_n) V_{\alpha_n}(\mathbf{Y}) \end{cases} \tag{46}$$

satisfy

$$C^* + h(\mathbf{X}) \leq \max_{a \in \mathbb{A}[\mathbf{X}]} \left(a r_j + \int h(\mathbf{Y}) P(d\mathbf{Y} | \mathbf{X}, a) \right). \tag{47}$$

Besides, the stationary policy obtained by maximizing the r.h.s. is optimal with expected average long-run profit equal to C^*. Finally, it turns out that (45) is actually stronger than Assumptions 3 and 4, so that where Ritt and Sennott apply Fatou's lemma in their proof of Theorem 2 we can apply instead the dominated convergence theorem, which allows to replace the inequality sign in (47) with an equality.

A.8. Proof of Proposition 4. Let $\alpha \in (0,1)$ and $\mathbf{a}^\alpha$ be an α-discount optimal policy. Starting with initial state $\mathbf{X}_0 = \mathbf{X}$, define policy $\mathbf{a}_\mathbf{X}$ through $a_\mathbf{X}(\mathbf{X}_k) = a^\alpha(\mathbf{X}'_k)$ for $k \geq 0$, where $\mathbf{X}'_k$ is the state that would have been obtained by following policy $\mathbf{a}^\alpha$ when faced with the exact same job arrival stream to date, but starting instead from initial state $\mathbf{X}'_0 = \mathbf{X}'$. From Lemma 2, Lemma 3 and Corollary 2 $W_k \leq W'_k$ for all $k \geq 0$, so from Theorem 1 the feasibility of policy $\mathbf{a}^\alpha$ implies that of policy $\mathbf{a}_\mathbf{X}$. In addition the revenue obtained with $\mathbf{a}_\mathbf{X}$ from the first job j when starting from $\mathbf{X}$ is larger than that of $\mathbf{a}^\alpha$ from j' starting from $\mathbf{X}'$ since $r_j \geq r_{j'}$, and for every arrival stream the discounted revenue streams of $\mathbf{a}_\mathbf{X}$ and $\mathbf{a}^\alpha$ are identical herafter, therefore

$$V_\alpha(\mathbf{a}_\mathbf{X}, \mathbf{X}) \geq V_\alpha(\mathbf{a}^\alpha, \mathbf{X}') \equiv V_\alpha(\mathbf{X}').$$

So $V_\alpha(\mathbf{a}_\mathbf{X}, \mathbf{X}) \leq V_\alpha(\mathbf{X})$ implies

$$V_\alpha(\mathbf{X}') \leq V_\alpha(\mathbf{X}), \tag{48}$$

and it follows from (48) that for any state $\mathbf{Y}$

$$V_\alpha(\mathbf{X}') - V_\alpha(\mathbf{Y}) \leq V_\alpha(\mathbf{X}) - V_\alpha(\mathbf{Y}),$$

which from (46) implies $h(\mathbf{X}') \leq h(\mathbf{X})$, completing the proof.

A.9. Proof of Proposition 5. From the Bellman equation (8) and the state dynamics (5), the optimal stationary policy $\mathbf{a} = (a(\cdot))_{k\geq 0}$ defined in Theorem 4 accepts a feasible job j while in state $\mathbf{X} = (W, j) \in \mathbb{X}$ if and only if

$$\begin{aligned} r_j \quad &> \sum_{\omega\in\mathbb{J}} \frac{\lambda_\omega}{\lambda} \int_0^{+\infty} [h(W[\tau],\omega) - h(W \cup j[\tau],\omega)]\, \lambda e^{-\lambda\tau} d\tau \\ &= \sum_{\omega\in\mathbb{J}} \frac{\lambda_\omega}{\lambda} \int_0^{\bar{d}} [h(W[\tau],\omega) - h(W \cup j[\tau],\omega)]\, \lambda e^{-\lambda\tau} d\tau, \end{aligned} \tag{49}$$

where the equality follows from the remark that $\tau \geq \bar{d}$ implies $W[\tau] = W \cup j[\tau] = 0$ for any W. But from (44) and (46) the differential value function satisfies for all $\mathbf{X} \in \mathbb{X}$

$$|h(\mathbf{X})| \leq \frac{\bar{r}}{\exp(-2\lambda\bar{d})}$$

so that the r.h.s of (49) can be bounded from above by

$$\sum_{\omega \in \mathbb{J}} \frac{\lambda_\omega}{\lambda} \int_0^{\bar{d}} \frac{2\bar{r}}{\exp(-2\lambda\bar{d})} \lambda e^{-\lambda\tau} d\tau = \frac{2\bar{r}(1-\exp(-\lambda\bar{d}))}{\exp(-2\lambda\bar{d})}. \tag{50}$$

It follows therefore from (49) and (50) that **a** will be myopic if

$$\begin{aligned} & \underline{r} > \frac{2\bar{r}(1-\exp(-\lambda\bar{d}))}{\exp(-2\lambda\bar{d})} \\ \iff \quad & \exp(-\lambda\bar{d}) > \frac{-\bar{r}+\sqrt{\bar{r}^2+2\bar{r}\underline{r}}}{\underline{r}}, \end{aligned}$$

where the equivalence stated follows from solving the previous quadratic inequality in $\exp(-\lambda\bar{d})$, which completes the proof.

A.10. Proof of Proposition 6. Let $\mathbf{a} = (a_k(\cdot))_{k\geq 1}$ be an optimal policy, and define random variable $Q_n^j(\mathbf{a})$ as the total amount of work from class j accepted by **a** during the first n arrivals, i.e.

$$Q_n^j(\mathbf{a}) \equiv q_j \sum_{\{k \in \{1,\ldots,n\}: j_k = j\}} a_k(\mathbf{X}_k). \tag{51}$$

Let now t_n be the time of the n-th arrival; the feasibility of **a** implies

$$\sum_{j \in \mathbb{J}} Q_n^j(\mathbf{a}) \leq t_n + \bar{d},$$

so it follows from the elementary renewal theorem that

$$\sum_{j \in \mathbb{J}} \liminf_{n \to +\infty} \frac{1}{n} E[Q_n^j(\mathbf{a})] \leq \frac{1}{\lambda}. \tag{52}$$

For all job class $j \in \mathbb{J}$ define now

$$\tilde{a}_j \equiv \frac{\lambda}{\lambda_j q_j} \liminf_{n \to +\infty} \frac{1}{n} E[Q_n^j(\mathbf{a})]; \tag{53}$$

notice that $\tilde{a}_j \leq 1$ since

$$\begin{aligned} \liminf_{n \to +\infty} \frac{1}{n} E[\frac{Q_n^j(\mathbf{a})}{q_j}] &\leq \liminf_{n \to +\infty} \frac{1}{n} E[\#\{k \in \{1,...,n\} : j_k = j\}] \\ &= \frac{\lambda_j}{\lambda}, \end{aligned}$$

where the inequality follows from (51) and the equality from basic properties of Poisson processes. In addition, substituting (53) in (52) yields

$$\sum_{j\in\mathbb{J}} \lambda_j q_j \tilde{a}_j \leq 1,$$

so that $(\tilde{a}_j)_{j\in\mathbb{J}}$ is a feasible solution to the problem (12) defining $\bar{C}^f$. Finally, observe that the r.v. defined as

$$R_n^j(\mathbf{a}) \equiv \frac{r_j}{q_j} Q_n^j(\mathbf{a})$$

represents the total profit from class j obtained by policy $\mathbf{a}$ during the first n job arrivals, so that from the definition (7)

$$\begin{aligned}
C^* &= C(\mathbf{a}) \\
&= \liminf_{n\to+\infty} \frac{1}{n} E[\sum_{j\in\mathbb{J}} R_n^j(\mathbf{a})] \\
&= \sum_{j\in\mathbb{J}} \frac{r_j}{q_j} \liminf_{n\to+\infty} \frac{1}{n} E[Q_n^j(\mathbf{a})] \\
&= \frac{1}{\lambda} \sum_{j\in\mathbb{J}} r_j \lambda_j \tilde{a}_j,
\end{aligned}$$

which implies $C^* \leq \bar{C}^f$ since the r.h.s of the last equality is the objective value of $(\tilde{a}_j)_{j\in\mathbb{J}}$ for the optimization problem (12), and completes the proof.

A.11. Proof of Theorem 5. This proof consists of three successive problem transformations eventually leading to the LP defining $LP_T(W)$. We describe each of these transformations as a separate proposition:

Proposition 10 *Define*

$$\begin{array}{rll}
G_T(W) \equiv & \max \sum_{j\in\mathbb{J}} \frac{r_j}{q_j} b_j(T+s_j) & \\
\textit{s.t.:} & 0 \leq \dot{b}_j(t) \leq \lambda_j q_j & (AD') \\
& \sum_{j=0}^{J} v_j(t) \leq 1 & (CP) \\
& b_j(t) \leq \int_0^t v_j(u)du \leq b_j(t+s_j) & (DA') \\
& W(t) \leq \int_0^t v_0(u)du & (DE) \\
& v_j(t) \geq 0 & (NG) \\
& b_j(t) = 0 \textit{ for } t \leq s_j. & (ST')
\end{array} \quad (54)$$

Then

$$F_T(W) = G_T(W).$$

Proof. Consider the following change of variables:

$$b_j(t) = \begin{cases} 0 \text{ if } t \in [0, s_j] \\ \lambda_j q_j \int_0^{t-s_j} a_j(u)du \text{ if } t \in [s_j, T+s_j] \end{cases}. \quad (55)$$

Note that the two objective expressions in (54) and (14) are equivalent, and that constraint (AD') in (54) is equivalent to constraint (FR) in (14). By integration, we see that constraint (BE) in (14) along with the initial condition $z_j(0) = 0$ in (MO) are equivalent to

$$z_j(t) = b_j(t + s_j) - \int_0^t v_j(u)du \text{ for all } j \text{ and } t \geq 0. \tag{56}$$

Substituting (56) in constraints (DA) and (MO) of (14) yields:

$$\int_0^t v_j(u)du \leq b_j(t + s_j) \leq \int_0^{t+s_j} v_j(u)du \text{ for all } j \geq 1 \text{ and } t \geq 0. \tag{57}$$

The second inequality in (57) is equivalent to the first inequality of (DA') in (54) because $v_j(t) \geq 0$ and $b_j(t) = 0$ for $t \leq s_j$, so (57) is equivalent to (DA'), which concludes the proof. ■

Proposition 11 *Define $(\tau_i)_{i\in\mathbb{I}}$ as in the statement of Theorem 5 and consider*

$$\begin{array}{lllr} GLP_T(W) & \equiv & \max\limits_{(V_i^j, B_i^j)} \sum_{j=1}^J \frac{r_j}{q_j} B_{T+s_j}^j & \\ & s.t.: & 0 \leq B_{i+1}^j - B_i^j \leq \lambda_j q_j(\tau_{i+1} - \tau_i) \text{ for all } (i,j) & (AD) \\ & & \sum_{j=0}^J (V_{i+1}^j - V_i^j) \leq \tau_{i+1} - \tau_i \text{ for all } (i,j) & (CP') \\ & & B_i^j \leq V_i^j \leq B_{i+s_j}^j \text{ for all } (i,j) & (DA'') \\ & & w_i \leq V_i^0 \text{ for all } i & (DE') \\ & & V_i^j \leq V_{i+1}^j \text{ for all } (i,j) & (NG') \\ & & B_i^j = 0 \text{ for all } i \text{ and } j \text{ such that } \tau_i \leq s_j & (ST) \end{array}, \tag{58}$$

where for notational simplicity $B_{i+s_j}^j \equiv B_{i'}^j$ where $i' \in \mathbb{I}$ is such that $\tau_{i'} = \tau_i + s_j$ – we know that such i' exists from the definition of $(\tau_i)_{i\in\mathbb{I}}$. Then

$$G_T(W) = GLP_T(W).$$

Proof. Let (V, B) be a feasible solution for (58), and for all j and $t \in [\tau_i, \tau_{i+1})$ define

$$\begin{cases} b_j(t) = \frac{\tau_{i+1}-t}{\tau_{i+1}-\tau_i} B_i^j + \frac{t-\tau_i}{\tau_{i+1}-\tau_i} B_{i+1}^j \\ v_j(t) = \frac{V_{i+1}^j - V_i^j}{\tau_{i+1}-\tau_i} \end{cases}. \tag{59}$$

Then $\dot{b}_j(t) = \frac{B_{i+1}^j - B_i^j}{t_{i+1}-t_i}$ for $t \in [\tau_i, \tau_{i+1})$, so that constraint (AD') of (54) is satisfied. $B_i^j \leq V_i^j$ and $B_{i+1}^j \leq V_{i+1}^j$ from constraint (DA'') imply

$$\begin{aligned}\int_0^t v_j(u)du &= \sum_{k=0}^{i-1}\left(V_{k+1}^j - V_k^j\right) + \frac{t-\tau_i}{\tau_{i+1}-\tau_i}(V_{i+1}^j - V_i^j)\\ &= \frac{\tau_{i+1}-t}{\tau_{i+1}-\tau_i}V_i^j + \frac{t-\tau_i}{\tau_{i+1}-\tau_i}V_{i+1}^j \qquad (60)\\ &\geq \frac{\tau_{i+1}-t}{\tau_{i+1}-\tau_i}B_i^j + \frac{t-\tau_i}{\tau_{i+1}-\tau_i}B_{i+1}^j\\ &= b_j(t), \qquad (61)\end{aligned}$$

and using $V_i^j \leq B_{i+s_j}^j$ implies likewise $\int_0^t v_j(u)du \leq b_j(t+s_j)$. Constraint (CP) of (54) is satisfied, since for $t \in [\tau_i, \tau_{i+1})$:

$$\begin{aligned}\sum_{j=0}^{J} v_j(t) &= \frac{1}{\tau_{i+1}-\tau_i}\sum_{j=0}^{J}\left(V_{i+1}^j - V_i^j\right)\\ &\leq 1, \qquad (62)\end{aligned}$$

where the inequality follows from constraint (CP') in (58). Finally,

$$\begin{aligned}\int_0^t v_0(u)du &= \frac{\tau_{i+1}-t}{\tau_{i+1}-\tau_i}V_i^0 + \frac{t-\tau_i}{\tau_{i+1}-\tau_i}V_{i+1}^0\\ &\geq \frac{\tau_{i+1}-t}{\tau_{i+1}-\tau_i}W(\tau_i) + \frac{t-\tau_i}{\tau_{i+1}-\tau_i}W(\tau_{i+1}) \qquad (63)\\ &= W(t), \qquad (64)\end{aligned}$$

where the first equality follows from the definition of $v_0(t)$, the inequality follows from constraint (DE') in (58), and the second equality from the fact that W is linear on $[\tau_i, \tau_{i+1})$ by construction of $(\tau_i)_{i\in\mathbb{I}}$. The last two constraints of (54) being obviously satisfied, we have thus proven that (b, v) is feasible for (54), which implies $G_T(W) \geq GLP_T(W)$ since (B, V) and (b, v) have the same objective value in their respective problems.

Let now (b, v) be a feasible solution for (54), and for all (i, j) define

$$\begin{cases} B_i^j = b_j(\tau_i) \\ V_i^j = \int_0^{\tau_i} v_j(u)du \end{cases}. \qquad (65)$$

Constraint (AD') in (54) implies $0 \leq \frac{b_j(\tau_{i+1})-b_j(\tau_i)}{\tau_{i+1}-\tau_i} \leq \lambda_j q_j$ for all (i, j), so that B satisfies the first constraint (AD) in (58). Constraint (DA') in (54) applied for $t = \tau_i$ implies that (B, V) satisfies constraint (DA'') in (58). Constraint (CP) implies $\int_{\tau_i}^{\tau_{i+1}} \sum_{j=0}^{J} v_j(u)du \leq$

$\tau_{i+1} - \tau_i$, which in turn implies constraint (CP') in (58). Finally, constraint (DE) of (54) applied for $t = t_i$ directly implies constraint (DE'). We have shown that (B, V) is feasible for problem (58) and therefore $GLP_T(W) \geq G_T(W)$, which concludes the proof. ■

Proposition 12

$$GLP_T(W) = LP_T(W).$$

Proof. Consider the increment variables $b_i^j = B_i^j - B_{i-1}^j$ and $v_i^j = V_i^j - V_{i-1}^j$. The LPs defining $GLP_T(W)$ and $LP_T(W)$ can be formulated instead as:

$$\begin{aligned} LP_T(W) \quad = \quad & \max_{(b_i^j)} \sum_{j=1}^{J} \frac{r_j}{q_j} \sum_{i=1}^{T+s_j} b_i^j \\ \text{s.t.:} \quad & w_i + \sum_{j=1}^{J} \sum_{k=1}^{i} b_i^j \leq t_i \text{ for all } i \\ & 0 \leq b_i^j \leq \mathbf{1}_{\{\tau_i > s_j\}} \lambda_j q_j (\tau_i - \tau_{i-1}) \text{ for all } (i,j) \end{aligned} \tag{66}$$

and

$$\begin{aligned} GLP_T(W) \quad = \quad & \max_{(b_i^j, v_i^j)} \sum_{j=1}^{J} \frac{r_j}{q_j} \sum_{i=1}^{T+s_j} b_i^j \\ \text{s.t.:} \quad & 0 \leq b_i^j \leq \mathbf{1}_{\{\tau_i > s_j\}} \lambda_j q_j (\tau_i - \tau_{i-1}) \text{ for all } (i,j) \\ & \sum_{k=1}^{i} b_k^j \leq \sum_{k=1}^{i} v_k^j \leq \sum_{k=1}^{i+s_j} b_k^j \text{ for all } (i,j) \\ & \sum_{j=0}^{J} v_i^j \leq \tau_i - \tau_{i-1} \text{ for all } i \\ & w_i \leq \sum_{k=1}^{i} v_k^0 \text{ for all } i \\ & v_i^j \geq 0 \text{ for all } (i,j) \end{aligned} \quad . \tag{67}$$

Let (b_i^j) any admission solution feasible for (66), and consider the problem of finding a scheduling solution (v_i^j) such that (b_i^j, v_i^j) is feasible for (67):

$$\begin{aligned} VLP(W) \quad = \quad & \min_{(v_i^j)} 0 \\ \text{s.t.:} \quad & \sum_{k=1}^{i} b_k^j \overset{(g_{ij})}{\leq} \sum_{k=1}^{i} v_k^j \overset{(e_{ij})}{\leq} \sum_{k=1}^{i+s_j} b_k^j \text{ for all } (i,j) \\ & \sum_{j=0}^{J} v_i^j \overset{(d_i)}{\leq} \tau_i - \tau_{i-1} \text{ for all } i \\ & w_i \overset{(h_i)}{\leq} \sum_{k=1}^{i} v_i^0 \text{ for all } i \\ & v_i^j \geq 0 \text{ for all } (i,j) \end{aligned} \quad . \tag{68}$$

The dual of problem (68) is:

$$\begin{aligned} VLP^*(W) \quad = \quad & \max_{(g_{ij},e_{ij},d_i,h_i)} \sum_{i=1}^{m} (h_i w_i - d_i(t_i - t_{i-1})) + \sum_{j=1}^{n}\sum_{i=1}^{m} \left(g_{ij} \sum_{k=1}^{i} b_k^j - e_{ij} \sum_{k=1}^{i+s_j} b_k^j \right) \\ \text{s.t.:} \quad & \sum_{k=i}^{m} h_k \leq d_i \text{ for all } i \\ & \sum_{k=i}^{m} (g_{kj} - e_{kj}) \leq d_i \text{ for all } (i,j) \\ & d_i, h_i, e_{ij}, g_{ij} \geq 0 \text{ for all } (i,j) \end{aligned} \tag{69}$$

The dual LP (69) is clearly feasible (with $(d,h,e,g) = 0$ as a solution) so according to duality theory, if by contradiction (68) were infeasible then (69) would necessarily be unbounded. This would imply the existence of an homogeneous solution with positive objective value, that is

$$\begin{aligned} & \sum_{k=i}^{m} h_k = d_i \text{ for all } i \\ & \sum_{k=i}^{m} (g_{kj} - e_{kj}) = d_i \text{ for all } (i,j) \\ & d_i, h_i, e_{ij}, g_{ij} \geq 0 \text{ for all } (i,j) \end{aligned} \tag{70}$$

and

$$\sum_{i=1}^{m} (h_i w_i - d_i(\tau_i - \tau_{i-1})) + \sum_{j=1}^{J}\sum_{i=1}^{m} \left(g_{ij} \sum_{k=1}^{i} b_k^j - e_{ij} \sum_{k=1}^{i+s_j} b_k^j \right) > 0 \ . \tag{71}$$

The two equalities in (70) imply $h_i = g_{ij} - e_{ij}$, so that the left-hand side of (71) would be equal to:

$$\begin{aligned} & \sum_{i=1}^{m} \left(h_i w_i - \sum_{k=i}^{m} h_k(\tau_i - \tau_{i-1}) \right) + \sum_{j=1}^{J}\sum_{i=1}^{m} \left((g_{ij} - e_{ij}) \sum_{k=1}^{i} b_k^j - e_{ij} \sum_{k=i+1}^{i+s_j} b_k^j \right) \\ = \; & \sum_{i=1}^{m} \left(h_i w_i - \sum_{k=i}^{m} h_k(\tau_i - \tau_{i-1}) \right) + \sum_{j=1}^{J}\sum_{i=1}^{m} \left(h_i \sum_{k=1}^{i} b_k^j - e_{ij} \sum_{k=i+1}^{i+s_j} b_k^j \right) \\ = \; & \sum_{i=1}^{m} h_i \left(w_i - \tau_i + \sum_{j=1}^{J}\sum_{k=1}^{i} b_k^j \right) - \sum_{j=1}^{J}\sum_{i=1}^{m} e_{ij} \sum_{k=i+1}^{i+s_j} b_k^j. \end{aligned}$$

Therefore (71) would imply

$$\sum_{i=1}^{m} h_i \left(w_i - \tau_i + \sum_{j=1}^{J}\sum_{k=1}^{i} b_k^j \right) > \sum_{j=1}^{J}\sum_{i=1}^{m} e_{ij} \sum_{k=i+1}^{i+s_j} b_k^j, \tag{72}$$

but since the first constraint of (66) implies $w_i - \tau_i + \sum_{j=1}^{J} \sum_{k=1}^{i} b_k^j \leq 0$ for all i, the left-hand side of (72) is necessarily non-positive while its right-hand side is necessarily non-negative, a contradiction. We have therefore proven that (68) is feasible, which implies that $GLP_T(W) \geq LP_T(W)$. On the other hand, the second, third and fourth constraints in (67) imply

$$\begin{array}{l} w_i \leq \sum_{k=1}^{i} v_k^0 \\ \leq \sum_{k=1}^{i} \left(\tau_k - \tau_{k-1} - \sum_{j=1}^{J} v_k^j \right) \\ \leq \tau_i - \sum_{j=1}^{J} \sum_{k=1}^{i} v_k^j \\ \leq \tau_i - \sum_{j=1}^{J} \sum_{k=1}^{i} b_k^j \end{array},$$

which imply the first constraint in (66), proves that $LP_T(W) \geq GLP_T(W)$, and completes the proof. ■

Acknowledgment. We would like to thank the participants of the OM seminar at the Stern School of Business (NYU), the joint OM/Marketing seminar at the Wharton School (UPenn), the OM Seminar at the Sloan School of Management (MIT), Dr. Maher Chebbo and the staff of SAP Inspire, and Dr. Timothy Kniker and the staff of Analytics Operations Engineering Inc. for useful discussions and feedback. This research project was partly funded by the Singapore-MIT alliance, J. Spencer Standish, Steve Graves, and the Doctoral Program of the MIT Sloan School of Management. We are particularly grateful to Steve Graves for both financial support and insightful remarks about this project.

References

Bertsekas, D. P. (1995). *Dynamic Programming and Optimal Control*, Volume 1. Belmont, MA: Athena Scientific.

Bertsekas, D. P. and J. N. Tsitsiklis (1996). *Neuro-Dynamic Programming*. Belmont, Massachusetts: Athena Scientific.

Dertouzos, M. L. (1974). Control robotics: The procedural control of physical processes. In *Proceedings of the IFIP Congress*, pp. 807–813.

Duenyas, I. (1995). Single facility due date setting with multiple customer classes. *Management Science 41*(4), 608–619.

Goldman, S. A., J. Parwatikar, and S. Suri (2000). On-line scheduling with hard deadlines. *Journal of Algorithms 34*(2), 370–389.

Goldwasser, M. H. (2003). Patience is a virtue: The effect of slack on competitiveness for admission control. *Journal of Scheduling*.

Kapuscinski, R. and S. Tayur (2003). Reliable due date setting in a capacitated MTO system with two customer classes. Technical report, University of Michigan.

Koren, G. and D. Shasha (1995). Dover: An optimal on-line scheduling algorithm for overloaded uniprocessor real-time systems. *SIAM Journal on Computing 24*(2), 318–339.

Lawler, E. L. (1990). A dynamic programming algorithm for preemptive scheduling of a single machine to minimize the number of late jobs. *Annals of Operations Research 26*, 125–133.

Locke, C. D. (1986). Best-effort decision making for real-time scheduling. Technical report, Carnegie Mellon University.

Maglaras, C. (2003). Revenue management for a multi-class single-server queue. Technical report, Columbia University.

Mieghem, J. V. (1995). Dynamic scheduling with convex delay costs: The generalized c-mu rule. *Annals of Applied Probability 5*, 809–833.

Plambeck, E. L. (2004). Optimal leadtime differentiation via diffusion approximations. *Operations Research 52*(2), 213–228.

Plambeck, E. L., S. Kumar, and M. J. Harrison (2001). Multiclass queue in heavy traffic with throughput time constraints: Asymptotically optimal dynamic controls. *Queueing Systems 39*, 23–54.

Ritt, R. K. and L. I. Sennott (1992). Optimal stationary policies in general state space markov decision chains with finite action sets. *Mathematics of Operations Research 17*(4), 901–909.

Talluri, K. T. and G. J. V. Ryzin (2004). *The Theory and Practice of Revenue Management.* Kluwer Academic Publishers.