MATHEMATICS OF OPERATIONS RESEARCH
Vol. 36, No. 2, May 2011, pp. 240–257
ISSN 0364-765X | EISSN 1526-5471 | 11 | 3602 | 0240

DOI 10.1287/moor.1110.0489
© 2011 INFORMS

Appointment Scheduling with Discrete Random Durations

Mehmet A. Begen
Ivey School of Business, University of Western Ontario, London, Ontario N6A 3K7, Canada,
mbegen@ivey.uwo.ca, http://www.ivey.uwo.ca/faculty/mbegen/

Maurice Queyranne
Sauder School of Business, University of British Columbia, Vancouver, British Columbia V6T 1Z2, Canada,
maurice.queyranne@sauder.ubc.ca

We consider the problem of determining an optimal appointment schedule for a given sequence of jobs (e.g., medical procedures) on a single processor (e.g., operating room, examination facility, physician), to minimize the expected total underage and overage costs when each job has a random processing duration given by a joint discrete probability distribution. Simple conditions on the cost rates imply that the objective function is submodular and L-convex. Then there exists an optimal appointment schedule that is integer and can be found in polynomial time. Our model can handle a given due date for the total processing (e.g., end of day for an operating room) after which overtime is incurred, as well as no-shows and some emergencies.

Key words: appointment scheduling; discrete convexity; optimization
MSC2000 subject classification: Primary: 90B36, 90C27; secondary: 90C15, 90C10
OR/MS subject classification: Primary: Health care; production/scheduling; inventory/production; secondary: programming, integer, stochastic; mathematics
History: Received June 15, 2009; revised March 22, 2010, February 8, 2011. Published online in *Articles in Advance* April 15, 2011.

1. Introduction and motivation. Our research concerns appointment scheduling of jobs on a highly utilized processor when the processing durations are stochastic, and jobs are not available before their appointment dates.[1] We came across this problem in surgery scheduling and in appointment scheduling of oncologist consultations and radiation therapy treatments for cancer patients. There are many other challenging and important real-life applications for this setting including healthcare diagnostic operations (such as CAT scans, MRIs) and physician appointments, as well as project scheduling, container vessel and terminal operations, gate and runway scheduling of aircrafts in an airport. For example, in surgery scheduling, patients or surgeries are the jobs, the operating room (OR) and associated resources are the processor, and the surgeon or the hospital is the scheduler. Figure 1 shows an example of surgery durations (OR time in minutes) per surgical specialty. This data was obtained during an applied research project (Santibanez et al. [32]). As seen from the box plots of Figure 1 and as reported in other studies (e.g., Begen et al. [5], Strum et al. [34]) surgery durations show variability.

Some appointment scheduling applications may have a specific due date for the end of processing, e.g., end-of-day for an OR, after which additional cost per time unit, e.g., overtime, is incurred. The need for a good schedule is crucial, and savings from such a schedule can be significant. In most cases, an appointment schedule needs to be prepared before any processing starts. It assigns each procedure an allocated duration by specifying the appointment date at which the required personnel and equipment, and the job or patient will be available. However, due to the uncertain processing durations, some jobs may finish earlier, whereas some others may finish later, than the appointment date of the next job. As the appointment dates have to be determined in advance, there are only limited recourse options when the actual duration of a job differs from its planned value. When a procedure finishes earlier than the next procedure's appointment date, the processor and other resources remain idle until the appointment date of the next job. This results in resource under-utilization. On the other hand, if a job finishes later than the next job's appointment date, the next job has to wait for the preceding procedure to complete and will start later than its original appointment date. This results in waiting for the next job and may cause overtime for the processor and resources at the end of the schedule. Therefore, there is an important tradeoff between underutilization, overtime, and job waiting times. We are interested in generating an appointment vector[2] that minimizes the expected total cost of resource under-utilization, overtime, and job waiting times. Finding such a schedule is more challenging but more valuable and useful when processing durations have more variability. Figure 2 shows an instance with three jobs G, B, R to be processed in this order. An appointment schedule (A_G, A_B, A_R) is given. Once the processing starts, due to the random processing durations, some jobs may be early whereas some others may be late as shown in Figure 2.

[1] To conform with scheduling terminology, we use the term "date" to denote a point in time. In most applications of appointment scheduling, the appointment "dates" are actually appointment *times* within the day for which the jobs are being scheduled.

[2] We use appointment schedule and appointment vector interchangeably.

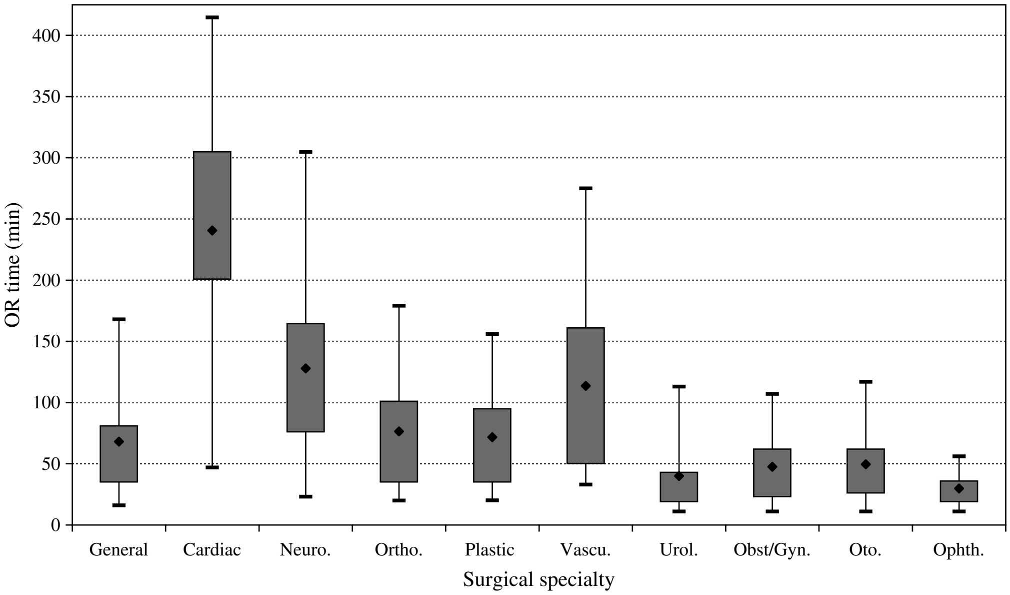

FIGURE 1. Surgery durations.

This problem can be modeled as a multistage stochastic program, but there are significant computational difficulties due to the need for multidimensional numerical integration (see §2). To the best of our knowledge, all the analytical studies we are aware of, even the ones with discrete epochs for job arrivals, use continuous processing duration distributions. For a given sequence of jobs, only small instances can be solved to optimality; larger instances require heuristics.

We study a discrete time version of the appointment scheduling problem and establish discrete convexity properties of the objective function. Discrete convex analysis has been advocated by Murota [22] and for recent developments in the topic see Murota [24]. We prove that the objective function is L-convex under mild assumptions on cost coefficients. L-convex functions, introduced by Murota [21], play a central role in discrete convexity and our research. Furthermore, we show that there exists an optimal *integer* appointment schedule minimizing expected total costs. This result is important as it allows us to optimize only over integer appointment schedules without loss of optimality. All these results on the objective function and optimal appointment schedule enable us to develop a polynomial time algorithm, based on discrete convexity (Murota [23]), that, for a given processing sequence, finds an appointment schedule minimizing the total expected cost. This algorithm invokes a sequence of submodular *set* function minimizations, for which various algorithms are available; see e.g., Fujishige [13], Iwata [17], Fleischer [12], McCormick [20], and Orlin [25].

When processing durations are stochastically independent, we evaluate the expected cost for a given processing order and an integer appointment schedule efficiently in polynomial time. Independent processing durations lead to faster algorithms.

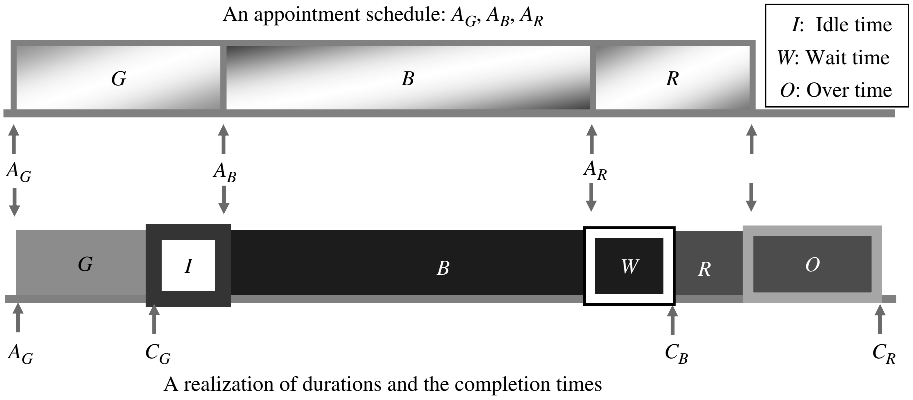

FIGURE 2. A three-job instance and a realization of the processing durations.

Our modeling framework can include a given due date for the end of processing (e.g., end of day for an operating room) after which overtime is incurred, instead of letting the model choose an end date. We also extend our analysis to include no-shows and some emergency jobs. The expected benefits of this research effort include reduced job-waiting times, reduced overtime, and improved capacity utilization.

Our paper is organized as follows. We start with a literature summary in §2. Section 3 states our assumptions, introduces our notation, and formally defines the problem and objective function. Section 4 gives some basic properties of the objective function and optimal solutions. In §5 we show the existence of an optimal appointment vector which is integer. Section 6 establishes the submodularity and L-convexity of the objective function under a mild condition on cost coefficients. We show that the total expected cost can be minimized efficiently and give the complexity of this minimization in §7. In that section, we also compute the objective function for any integer appointment vector and determine its complexity when the processing durations are stochastically independent. This independence assumption leads to faster algorithms. We extend our analysis for an objective function with a due date for the end of processing in §8. Section 9 shows how to handle no-shows and some emergency jobs within our framework. Section 10 discusses the current work and future work, and it concludes the paper.

2. Related literature. There are many studies in the last 50 years about appointment scheduling, especially in healthcare. Here, we present the ones that we believe are the most relevant to our research. The use of appointment systems is not limited to service industries but also extends to other areas, such as project management, production, and transportation.

Sabria and Daganzo [31] consider scheduling of arrivals of container vessels at a seaport. Weiss [38] recognized that the appointment scheduling problem has a closed form solution when there is only a single job, and it coincides with the well-known newsvendor solution from inventory theory. However, the appointment scheduling problem departs from newsvendor characteristics and solution methods in the case of multiple jobs (Robinson et al. [30], Begen and Queyranne [3]). Zipkin [39] presents an analysis on the structure of a single-item multiperiod inventory system, closely related to the newsvendor problem, by using discrete convexity. Elhafsi [11] studies a production system of multiple stages with stochastic leadtimes. The objective is to determine planned leadtimes such that the expected total cost (inventory, tardiness, and earliness) is minimized. Bendavid and Golany [6] consider project scheduling with stochastic activity durations. They address the problem of determining for each activity a *gate*, i.e., a time before which the activity cannot begin, so as to minimize total expected holding and shortage costs, for which they use a heuristic based on the cross entropy methodology. Cayirli and Veral [9] review the literature on appointment systems of outpatient scheduling. The authors classify the literature in terms of methodologies and modeling aspects considered, and provide a discussion of performance measures. The authors conclude that the existing literature provides very situation-specific solutions and does not offer generally applicable and portable methodologies for appointment systems design in outpatient scheduling. Another literature review by Cardoen et al. [8] on operating room scheduling evaluates the papers on either the problem setting, such as performance measures, or technical properties such as solution methods.

In a queuing-based study, Wang [36] develops a model to find appointment dates of jobs in a single-server system to minimize expected customer delay and server completion time with identical jobs, identical costs, and exponentially distributed processing durations. In his numerical studies, the optimal allocated time for each job shows a “dome” structure; i.e., it increases first and then decreases. In another study, Wang [37] investigates the sequencing problem with the same setting but with distinct exponential distributions. He conjectures that sequencing with increasing variance is optimal. Bosch et al. [7] present a model with i.i.d. Erlang processing durations and identical cost coefficients. In their model, customers can arrive only at one of discrete potential arrival epochs, which are equally spaced, and the decision variable is the number of customers to be scheduled at each potential arrival epoch. In a related paper, Kaandorp and Koole [18] study outpatient appointment scheduling with exponential processing durations and no-shows. They take advantage of the exponential distribution in their computations and define a neighborhood structure and an exact search method. However, for large instances, they develop a heuristic due to high computation times of their search method.

Another important stream of appointment scheduling research is based on stochastic programming. Denton and Gupta [10] develop a two-stage stochastic linear program to determine optimal appointment dates for a given surgery sequence and due date for the end-of-processing horizon. The authors use general, i.i.d., and continuous processing durations, and identical server idling cost coefficients for all jobs. They infer from stochastic programming results that their model is a convex minimization problem, and they develop an algorithm with sequential bounding for solving small-sized instances. They develop heuristics to solve larger instances. In a related study, Robinson and Chen [29] develop a stochastic linear program for finding appointment dates for a fixed sequence of surgeries and propose a Monte Carlo-based solution method. Due to the high computational requirements of Monte Carlo integration, they develop heuristics in which they use the "dome" structure of the optimal policy as reported in Wang [36].

Appointment scheduling can be thought of as an operational level of capacity planning problem since it is concerned with scheduling of jobs (patients) available on the day of processing (service) (Santibanez et al. [32], Patrick et al. [26], Schutz and Kolisch [33]). Researchers also study the problem of scheduling patients in advance of the service date. In this stream of research see, e.g., Patrick et al. [26], Schutz and Kolisch [33], Green et al. [14], Gupta and Wang [15], and the references therein, arrivals are random but processing durations are deterministic and the main decision is how to allocate available capacity to incoming demand. Different objectives are considered such as revenue maximization (Gupta and Wang [15]) or cost minimization to achieve target waiting times (Patrick et al. [26]). Luzon et al. [19] use a fluid approximation to minimize average waiting time.

Finally, we would like to point out the similarities between appointment scheduling and single machine scheduling; see e.g., Pinedo [28] for machine scheduling. Unlike machine scheduling, in appointment scheduling a sequence is given and the release dates are the decision variables. Furthermore, the objective function of the appointment scheduling problem is quite different than the objective functions of classical machine scheduling problems. Processing durations are usually deterministic in machine scheduling problems but random processing durations are also studied in literature; see, e.g., Pinedo [27, 28].

In this paper we develop a framework for solving appointment scheduling problems with discrete duration distributions. The framework neither requires identical and independently distributed durations nor identical cost coefficients. (We do require a mild condition on cost coefficients.) Finally the framework can handle no-shows and some types of emergencies. We believe the framework is portable and applicable to many appointment systems in healthcare and other areas.

3. Assumptions and notation.

There are $n+1$ jobs that need to be sequentially processed on a single processor. The processing sequence is given. An appointment schedule needs to be prepared before any processing can start and jobs will not be available before their appointment dates. When a job finishes earlier than the next job's appointment date, the system experiences some cost due to under-utilization. We refer to this cost as the *underage* cost. On the other hand, if a job finishes later than the next job's appointment date, the system experiences *overage* cost due to the overtime of the current job and the waiting of the next job.

The processing durations are given by their joint discrete distribution. In §7, we will show that assuming independent discrete processing durations lead to faster algorithms. We assume that this joint distribution is known. Complete information of distributions is reasonable in most settings, but we relax this assumption in Begen et al. [4]. Our next assumption is a natural one: all cost coefficients and processing durations are nonnegative and bounded. A key assumption in this work is that processing durations are integer-valued.[3] Although we obtain some of our results without this assumption, it is important for our main results.

We assume job 1 starts on time; i.e., the start time for the first job is zero, and there are n real jobs. The $(n+1)$st job is a dummy job with a processing duration of 0. The appointment time for the $(n+1)$st job is the total time available for the n real jobs. We use the dummy job to compute the overage or underage cost of the nth job.

Let $\{1, 2, 3, \ldots, n, n+1\}$ denote the set of jobs. We denote the random processing duration of job i by p_i and the random vector[4] of processing durations by $\mathbf{p} = (p_1, p_2, \ldots, p_n, 0)$. Let $\underline{p}_i$ and $\bar{p}_i$ denote the minimum and maximum possible value of processing duration p_i, respectively. The maximum of these $\bar{p}_i$s is $\bar{p}_{\max} = \max(\bar{p}_1, \ldots, \bar{p}_n)$. The *underage cost rate* u_i of job i is the unit cost (per time unit) incurred when job i is completed at a date C_i before the appointment date A_{i+1} of the next job $i+1$. The *overage cost rate* o_i of job i is the unit cost incurred when job i is completed at a date C_i after the appointment date A_{i+1}. Thus the total cost due to job i completing at date C_i is $u_i(A_{i+1} - C_i)^+ + o_i(C_i - A_{i+1})^+$ where $(x)^+ = \max(0, x)$ is the positive part of real number x. We define $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{o} = (o_1, o_2, \ldots, o_n)$. We denote unit vectors in $\mathbb{R}^{n+1}$ as $\mathbf{1}_i$ where the ith component is 1 and all other components are 0.

The underage cost may be interpreted as the idling cost and/or opportunity cost of the resources, whereas the overage cost may be thought of as the waiting cost of the next job and/or the overtime of the current job. The overage cost of the last job may include the overtime cost for the whole facility at the end of the schedule after a specified due date.

[3] We can restrict ourselves to integer appointment schedules without loss of optimality by Theorem 5.1.

[4] We write all vectors as row vectors.

Next we introduce our decision variables. Let a_i be the allocated duration and let A_i be the appointment date for job i. Then we have $A_1 = 0$ and $A_{i+1} = A_i + a_i$ for $i = 1, \ldots, n$. Thus we may equivalently use the *allocated duration vector* $\mathbf{a} = (a_1, a_2, \ldots, a_{n-1}, a_n)$ or the *appointment vector* $\mathbf{A} = (A_1, A_2, \ldots, A_n, A_{n+1})$ (with $A_1 = 0$) as our decision variables; we choose to work with the appointment vector $\mathbf{A}$. We introduce additional variables that help define and compute the objective function. Let S_i be the start date and let C_i be the completion date of job i. Since job 1 starts on time, we have $S_1 = 0$ and $C_1 = p_1$. The other start times and completion times are determined as follows: $S_i = \max\{A_i, C_{i-1}\}$ and $C_i = S_i + p_i$ for $2 \le i \le n+1$. Note that the dates S_i and C_i are random variables that depend on the appointment vector $\mathbf{A}$.

Let $F(\mathbf{A} \mid \mathbf{p})$ be the total cost of appointment vector $\mathbf{A}$ given processing duration vector $\mathbf{p}$:

$$F(\mathbf{A} \mid \mathbf{p}) = \sum_{i=1}^{n} \big(o_i (C_i - A_{i+1})^+ + u_i (A_{i+1} - C_i)^+ \big). \tag{1}$$

The objective to be minimized is the expected total cost $F(\mathbf{A}) = \mathrm{E}_p[F(\mathbf{A} \mid \mathbf{p})]$, where the expectation is taken with respect to random processing duration vector $\mathbf{p}$. We simplify notations by defining the lateness $L_i = C_i - A_{i+1}$ of job i, its tardiness $T_i = (L_i)^+$, and its earliness $E_i = (-L_i)^+$. The objective $F(\mathbf{A})$ can now be written as

$$F(\mathbf{A}) = \mathrm{E}_p\left[\sum_{i=1}^{n} (o_i T_i + u_i E_i) \right] = \sum_{i=1}^{n} (o_i \mathrm{E}_p T_i + u_i \mathrm{E}_p E_i).$$

4. Basic properties. We start by making an observation about the completion times and expressing the objective function in a different form that is useful for deriving some of our later results. Since $C_i = S_i + p_i = \max\{A_i, C_{i-1}\} + p_i$, the completion time of job i may be seen as the length of the longest (or critical) path from some job j $(j \le i)$ to job $i+1$ in a corresponding "project network" (Pinedo [28]), namely:

Lemma 4.1 (Critical Path). *For all jobs $i = 1, \ldots, n$,*

$$C_i = \max_{j \le i} \left\{ A_j + \sum_{k=j}^{i} p_k \right\},$$

$$F(\mathbf{A} \mid \mathbf{p}) = \sum_{i=1}^{n} \left(o_i \left(\max_{j \le i} \left\{ A_j + \sum_{k=j}^{i} p_k \right\} - A_{i+1} \right)^+ + u_i \left(A_{i+1} - \max_{j \le i} \left\{ A_j + \sum_{k=j}^{i} p_k \right\} \right)^+ \right).$$

Proof. The claim holds trivially for $i = 1$. By induction let the claim be true for $i = m$; i.e., $C_m = \max_{j \le m}\{A_j + \sum_{k=j}^{m} p_k\}$. Then

$$\begin{aligned} C_{m+1} &= S_{m+1} + p_{m+1} = \max\{A_{m+1}, C_m\} + p_{m+1} \quad \text{by definition} \\ &= \max\left\{ A_{m+1}, \max_{j \le m} \left\{ A_j + \sum_{k=j}^{m} p_k \right\} \right\} + p_{m+1} \quad \text{by inductive assumption} \\ &= \max\left\{ A_{m+1} + p_{m+1}, \max_{j \le m} \left\{ A_j + \sum_{k=j}^{m+1} p_k \right\} \right\} = \max_{j \le m+1} \left\{ A_j + \sum_{k=j}^{m+1} p_k \right\}. \end{aligned}$$

The expression for $F(\mathbf{A} \mid \mathbf{p})$ follows. □

The next result is not only important on its own but also crucial for the existence of an optimal solution.

Lemma 4.2 (Continuity). *Functions $F(\cdot \mid \mathbf{p})$ and $F(\cdot)$ are continuous.*

Proof. By Equation (1), $F(\cdot \mid \mathbf{p})$ is a weighted sum of piecewise linear continuous functions of $\mathbf{A}$, hence is itself piecewise linear continuous. Since we have a finite sample space, the expectation $F(\cdot) = \mathrm{E}_p F(\cdot \mid \mathbf{p})$ is also continuous. □

We next establish the existence of an optimal solution. The proof follows from the fact that our objective function is continuous (by Lemma 4.2), and we can restrict the appointment vector to a compact set without loss of optimality. Let $\underline{\mathbf{A}} = (\underline{A}_1, \ldots, \underline{A}_{n+1})$ and $\bar{\mathbf{A}} = (\bar{A}_1, \ldots, \bar{A}_{n+1})$, where $\underline{A}_1 = \bar{A}_1 = 0$, $\underline{A}_i = \sum_{j<i} \underline{p}_j$, and $\bar{A}_i = \sum_{j<i} \bar{p}_j$ for $i = 2, \ldots, n+1$. We define the compact set $\mathscr{K}$ as the cartesian product of the intervals $[\underline{A}_i, \bar{A}_i]$; i.e., $\mathscr{K} = \prod_{i=1}^{n+1} [\underline{A}_i, \bar{A}_i] = [\underline{\mathbf{A}}, \bar{\mathbf{A}}] \subseteq \mathbb{R}^{n+1}$.

Lemma 4.3 (Existence of an Optimal Vector). *There exists an appointment vector $\mathbf{A}^* \in \mathscr{K}$ such that $F(\mathbf{A}^*) \le F(\mathbf{A})$ for any appointment vector $\mathbf{A}$.*

PROOF. We show that we can restrict, without loss of optimality, the appointment vector $\mathbf{A}$ to the compact set $\mathscr{K} = [\underline{\mathbf{A}}, \bar{\mathbf{A}}]$ and recall that job 1 starts at time zero; i.e., $A_1 = 0 = \underline{A}_1 = \bar{A}_1$. Consider any appointment vector $\mathbf{A} \notin \mathscr{K}$ with $A_1 = 0$.

If $\mathbf{A} \not\geq \underline{\mathbf{A}}$, then define the appointment vector $\mathbf{A}' = \mathbf{A} \vee \underline{\mathbf{A}}$ with component $A'_i = \max\{A_i, \underline{A}_i\}$. For any realization $\mathbf{p}$ of the processing durations, the completion times C'_i in the resulting schedule satisfy $C'_i = C_i \geq \underline{A}_{i+1}$. (Indeed, $C'_1 = p_1 = C_1 \geq \underline{A}_2$ and, by induction, $C'_i = \max\{A'_i, C'_{i-1}\} + p_i = \max\{A_i, \underline{A}_i, C_{i-1}\} + p_i = \max\{A_i, C_{i-1}\} + p_i = C_i \geq \underline{A}_{i+1}$.) Then the resulting tardiness and earliness become: if $A_{i+1} \geq \underline{A}_{i+1}$, then $T'_i = (C'_i - A'_{i+1})^+ = (C_i - A_{i+1})^+ = T_i$ and $E'_i = (A'_{i+1} - C'_i)^+ = (A_{i+1} - C_i)^+ = E_i$; and, if $A_{i+1} < \underline{A}_{i+1}$, then $T'_i = (C'_i - A'_{i+1})^+ = (C_i - \underline{A}_{i+1})^+ \leq (C_i - A_{i+1})^+ = T_i$ and $0 \leq E_i = (A_{i+1} - C_i)^+ \leq (A'_{i+1} - C'_i)^+ = E'_i = 0$ (so $E'_i = E_i = 0$). Since all $u_i, o_i \geq 0$, it follows from Equation (1) that $F(\mathbf{A}' \mid \mathbf{p}) \leq F(\mathbf{A} \mid \mathbf{p})$ and thus, $F(\mathbf{A}') \leq F(\mathbf{A})$. We have shown that for every $\mathbf{A}$ there exists $\mathbf{A}' \geq \underline{\mathbf{A}}$ with $F(\mathbf{A}') \leq F(\mathbf{A})$.

Now, for any vector $\mathbf{A} \in \mathbb{R}^{n+1}$ satisfying $\mathbf{A} \geq \underline{\mathbf{A}}$, $A_1 = 0$, and $\mathbf{A} \notin \mathscr{K}$, let $i(\mathbf{A})$ denote the smallest index such that $A_i > \bar{A}_i$. Let $\mathbf{A} \in \mathbb{R}^{n+1}$ be a vector with largest $i(\mathbf{A})$ value satisfying $\mathbf{A} \geq \underline{\mathbf{A}}$, $A_1 = 0$, and $\mathbf{A} \notin \mathscr{K}$. We claim that there exists $\mathbf{A}'$ satisfying $\mathbf{A}' \geq \underline{\mathbf{A}}$, $A'_1 = 0$, $F(\mathbf{A}' \leq F(\mathbf{A})$, and either $\mathbf{A}' \in \mathscr{K}$ or $i(\mathbf{A}') > i(\mathbf{A})$. Then after at most n such changes we obtain $\mathbf{A}'' \in \mathscr{K}$ satisfying $F(\mathbf{A}'') \leq F(\mathbf{A})$, which is what we wanted to show. We now prove the claim.

Let $i = i(\mathbf{A})$, $\varepsilon = A_i - \bar{A}_i > 0$, and define $\mathbf{A}'$ with $A'_j = A_j$ for all $j \leq i-1$, and $A'_j = A_j - \varepsilon$ for all $j \geq i$, so $A'_i = \bar{A}_i$. For every realization $\mathbf{p}$ of the processing durations, the completion times C'_j in the resulting schedule satisfy $C'_j = C_j$ for all $j \leq i-1$. Note that for all $j \leq i-1$, $A_j \leq \bar{A}_j$ implies $C_j \leq \bar{A}_{j+1}$. Therefore, $C_i = A_i + p_i$ and $C'_i = A'_i + p_i = A'_i + C_i - A_i = C_i - \varepsilon$. It follows that $C'_j = C_j - \varepsilon$ for all $j \geq i$. As a result, $E'_j = E_j$ and $T'_j = T_j$ for all $j \neq i-1$, and $E'_{i-1} = E_{i-1} - \varepsilon$, $T'_{i-1} = T_{i-1} = 0$. Since $\varepsilon > 0$ and $u_{i-1} \geq 0$, $F(\mathbf{A}' \mid \mathbf{p}) \leq F(\mathbf{A} \mid \mathbf{p})$ and thus, $F(\mathbf{A}') \leq F(\mathbf{A})$. Since $A'_j = A_j \leq \bar{A}_j$ for all $j \leq i-1$ and $A'_i = \bar{A}_i$, then either $\mathbf{A}' \in \mathscr{K}$ or $i(\mathbf{A}') \geq i+1 = i(\mathbf{A})+1$, establishing the claim. This shows that for any $\mathbf{A} \notin \mathscr{K}$ there exists a vector $\mathbf{A}'' \in \mathscr{K}$ with $F(\mathbf{A}'') \leq F(\mathbf{A})$.

As a result, since F is continuous, its minimum on compact set $\mathscr{K}$ is attained and is therefore the global minimum. □

The next lemma gives bounds on the difference between any two consecutive components of an optimal appointment vector, and from this we obtain a useful and intuitive result in Lemma 4.5.

LEMMA 4.4. *There exists an optimal appointment schedule* $\mathbf{A}^* \in \mathscr{K}$ *satisfying* $\underline{p}_i \leq A^*_{i+1} - A^*_i \leq \sum_{j \leq i} \bar{p}_j - \sum_{j<i} \underline{p}_j$ *for all* $i = 1, \ldots, n$.

PROOF. By Lemma 4.3, we immediately obtain $\underline{p}_1 \leq A^*_2 - A^*_1 \leq \bar{p}_1$ and $A^*_{i+1} - A^*_i \leq \sum_{j \leq i} \bar{p}_j - \sum_{j<i} \underline{p}_j$ for all $i = 2, \ldots, n$. Next, we show that $\underline{p}_i \leq A^*_{i+1} - A^*_i$ holds for all $i = 2, \ldots, n$. For contradiction, suppose $\underline{p}_k + A^*_k > A^*_{k+1}$ for some $k = 2, \ldots, n$. Then job k is late at least $(\underline{p}_k + A^*_k - A^*_{k+1})$ time units, so increasing A^*_{k+1} to $\underline{p}_k + A^*_k$ will improve the objective function by $o_k(\underline{p}_k + A^*_k - A^*_{k+1}) \geq 0$. Therefore, we must have $\underline{p}_i \leq A^*_{i+1} - A^*_i$ for all $i = 2, \ldots, n$. □

LEMMA 4.5 (NONDECREASING APPOINTMENT DATES). *There exists an optimal appointment vector* $\mathbf{A}^* \in \mathscr{K}$ *with nondecreasing components*; *i.e.,* $A^*_i \leq A^*_{i+1}$ *for all* $i = 1, \ldots, n$.

PROOF. By Lemma 4.4, $A^*_{i+1} - A^*_i \geq \underline{p}_i \geq 0$ $(1 \leq i \leq n)$. □

5. Optimality of an integer appointment vector.

The existence of an optimal appointment vector that is integer is crucial. It implies that we can restrict attention to integer appointment vectors without loss of optimality. We establish this result in the appointment vector integrality Theorem 5.1. Its proof is surprisingly nontrivial.

Let $\mathbf{A}^*$ be any noninteger appointment vector and let A^*_f be the first noninteger component of $\mathbf{A}^*$. Knowing all the jobs which have the same fractional part as A^*_f is crucial, so we define J to be the set of all jobs $j \geq f$ such that $A^*_j - A^*_f$ is integer. Let $\mathbb{Z}$ denote the set of integers, and let $\lfloor x \rfloor = \sup\{n \in \mathbb{Z}\colon n \leq x\}$ and $\lceil x \rceil = \inf\{n \in \mathbb{Z}\colon n \geq x\}$ for $x \in \mathbb{R}$. Let $\varphi(x)$ be the distance to the nearest integer for $x \in \mathbb{R}$; i.e., $\varphi(x) = \min(x - \lfloor x \rfloor, \lceil x \rceil - x)$. Let Δ be a strictly positive scalar satisfying $0 < \Delta < \frac{1}{2}\min(\Delta_1, \Delta_2)$, where $\Delta_1 = \frac{1}{4}\min\{\varphi(|A^*_j - A^*_k|)\colon j \in J, k \notin J\} > 0$ and $\Delta_2 = \frac{1}{4}\min\{\varphi(|A^*_j - A^*_k|)\colon j \notin J, k \notin J, A_j - A_k \notin \mathbb{Z}\} > 0$. We use Δ to construct two new appointment schedules $\mathbf{A}'$ and $\mathbf{A}''$ from $\mathbf{A}^*$: let $A'_j = A^*_j - \Delta$ if $j \in J$, and $A'_j = A^*_j$ otherwise; similarly, let $A''_j = A^*_j + \Delta$ if $j \in J$, and $A''_j = A^*_j$ otherwise. For any realization of the processing duration vector $\mathbf{p}$, denote the completion times of job j as C^*_j, C'_j, C''_j in schedules $\mathbf{A}^*$, $\mathbf{A}'$, and $\mathbf{A}''$, respectively.

One of the main ideas in proving the appointment vector integrality Theorem 5.1 is that Δ is small enough so that there is "no event change" when we move from schedule $\mathbf{A}^*$ to schedules $\mathbf{A}'$ and $\mathbf{A}''$. When there is no event change, we show in Lemma 5.9 that our objective function changes linearly between schedules $\mathbf{A}'$

and $\mathbf{A}''$. To make the no-event change concept precise, we define the following. Job i $(1 < i \leq n+1)$ is *late* if $C^*_{i-1} > A^*_i$ (strictly positive tardiness), *early* if $C^*_{i-1} < A^*_i$ (strictly positive earliness), *just-on-time* if $C^*_{i-1} = A^*_i$, and *on-time* if $C^*_{i-1} \leq A^*_i$. Then, no-event change means that if any job is late, early, or just-on-time, respectively, in schedule $\mathbf{A}^*$, then it is also late, early, or just-on-time, respectively, in both schedules $\mathbf{A}'$ and $\mathbf{A}''$.

We consider all possible realizations $\mathbf{r}$ of the random processing duration vector $\mathbf{p}$, so r_i is the corresponding realization of the processing duration p_i. We start by establishing relationships between the completion times in the schedules $\mathbf{A}'$ and $\mathbf{A}^*$, and $\mathbf{A}''$ and $\mathbf{A}^*$.

Lemma 5.1. *For every realization of the processing durations and every $j = 1, \ldots, n+1$, $C^*_j + \Delta \geq C''_j \geq C^*_j \geq C'_j \geq C^*_j - \Delta$.*

Proof. Let $1 \leq j \leq n+1$ and let $\mathbf{r}$ be a realization of $\mathbf{p}$. Then $A^*_j - \Delta \leq A'_j \leq A^*_j \leq A''_j \leq A^*_j + \Delta$ by definition of $\mathbf{A}'$ and $\mathbf{A}''$. By the critical path Lemma 4.1, $C^*_j = \max_{k \leq j}\{A^*_k + \sum_{i=k}^{j} r_i\}$, $C'_j = \max_{k \leq j}\{A'_k + \sum_{i=k}^{j} r_i\}$, and $C''_j = \max_{k \leq j}\{A''_k + \sum_{i=k}^{j} r_i\}$. Hence, $A'_j \leq A^*_j \leq A''_j$ implies that $C'_j \leq C^*_j \leq C''_j$. On the other hand, $A^*_j - \Delta \leq A'_j$ implies that $C^*_j - \Delta = \max_{k \leq j}\{A^*_k - \Delta + \sum_{i=k}^{j} r_i\} \leq \max_{k \leq j}\{A'_k + \sum_{i=k}^{j} r_i\} = C'_j$, so $C^*_j - \Delta \leq C'_j$. Similarly $A^*_j + \Delta \geq A''_j$ implies that $C^*_j + \Delta = \max_{h \leq j}\{A^*_k + \Delta + \sum_{i=k}^{j} r_i\} \geq \max_{k \leq j}\{A''_k + \sum_{i=k}^{j} r_i\} = C''_j$, so $C^*_j + \Delta \geq C''_j$. The result follows. □

The next two results are about late and early jobs. Lemma 5.2 below implies that if job k is late (resp., early), then its tardiness (resp., earliness) is strictly greater than 2Δ. Lemma 5.3 implies that if job k is late (resp., early) in schedule $\mathbf{A}^*$, then it is also late (resp., early) in $\mathbf{A}'$ and $\mathbf{A}''$.

Lemma 5.2. *For every realization of the processing durations and every $k = 2, \ldots, n+1$, if $|C^*_{k-1} - A^*_k| > 0$, then $|C^*_{k-1} - A^*_k| > 2\Delta$.*

Proof. Let $\mathbf{r}$ be a realization of $\mathbf{p}$. Let t be the last on-time job before k $(1 \leq t < k)$ so $C^*_{k-1} = A^*_t + \sum_{i=t}^{k-1} r_i$. Note that t exists and is well defined since job 1 is always on time; i.e., $A^*_1 = 0$. We consider two cases: $(A^*_k - A^*_t) \in \mathbb{Z}$ or $(A^*_k - A^*_t) \notin \mathbb{Z}$. If $(A^*_k - A^*_t) \in \mathbb{Z}$, then $0 < |C^*_{k-1} - A^*_k| = |A^*_t + \sum_{i=t}^{k-1} r_i - A^*_k|$, but since the r_is and $A^*_k - A^*_t$ are integer, $|A^*_t + \sum_{i=t}^{k-1} r_i - A^*_k|$ is a positive integer, hence $|C^*_{k-1} - A^*_k| = |A^*_t + \sum_{i=t}^{k-1} r_i - A^*_k| \geq 1 > 2\Delta$. If $A^*_k - A^*_t$ is not integer, then $\varphi(A^*_k - A^*_t) > 2\Delta$, and this implies $\lceil A^*_k - A^*_t \rceil - (A^*_k - A^*_t) > 2\Delta$ and $(A^*_k - A^*_t) - \lfloor A^*_k - A^*_t \rfloor > 2\Delta$. Since $\sum_{i=t}^{k-1} r_i$ is integer, we must have either $\sum_{i=t}^{k-1} r_i \leq \lfloor A^*_k - A^*_t \rfloor$ or $\sum_{i=t}^{k-1} r_i \geq \lceil A^*_k - A^*_t \rceil$. Therefore $|C^*_{k-1} - A^*_k| = |A^*_t + \sum_{i=t}^{k-1} r_i - A^*_k| > 2\Delta$. □

Lemma 5.3. *For every realization of the processing duration and every $k = 2, \ldots, n+1$, if $C^*_{k-1} > A^*_k$, then $C'_{k-1} > A'_k$ and $C''_{k-1} > A''_k$, and if $C^*_{k-1} < A^*_k$, then $C'_{k-1} < A'_k$ and $C''_{k-1} < A''_k$.*

Proof. By Lemma 5.2, $C^*_{k-1} > A^*_k$ implies $C^*_{k-1} - A^*_k > 2\Delta$. Note that $A^*_k - \Delta \leq A'_k \leq A^*_k \leq A''_k \leq A^*_k + \Delta$ by definition, and $C^*_{k-1} + \Delta \geq C''_{k-1} \geq C^*_{k-1} \geq C'_{k-1} \geq C^*_{k-1} - \Delta$ by Lemma 5.1. Then $C^*_{k-1} - A^*_k > 2\Delta$ implies $C'_{k-1} - A'_k \geq C^*_{k-1} - \Delta - A^*_k > \Delta$ and $C''_{k-1} - A''_k \geq C^*_{k-1} - A^*_k - \Delta > \Delta$. Similarly, by Lemma 5.2, $C^*_{k-1} < A^*_k$ implies $A^*_k - C^*_{k-1} > 2\Delta$. Note that $A^*_k - \Delta \leq A'_k \leq A^*_k \leq A''_k \leq A^*_k + \Delta$ by definition, and $C^*_{k-1} + \Delta \geq C''_{k-1} \geq C^*_{k-1} \geq C'_{k-1} \geq C^*_{k-1} - \Delta$ by Lemma 5.1. Then $A^*_k - C^*_{k-1} > 2\Delta$ implies $A'_k - C'_{k-1} \geq A^*_k - C^*_{k-1} - \Delta > \Delta$ and $A''_k - C''_{k-1} \geq A^*_k - C^*_{k-1} - \Delta > \Delta$. The result follows. □

Just-on-time jobs require more care, and we need further definitions and results before we can establish similar results as in Lemmas 5.2 and 5.3. Let a *block* $B[t, k]$ be a sequence of consecutive jobs, $[t, t+1, \ldots, k]$ $(1 \leq t < k \leq n+1)$, such that either $t = 1$ or job t is early, i.e., $C^*_{t-1} < S^*_t = A^*_t$; no other job in the block is early, i.e., $S^*_{j+1} = C^*_j \geq A^*_{j+1}$ for $j = t+1, \ldots, k$; and job k is just-on-time, i.e., $C^*_{k-1} = A^*_k$. Let $K = \{i\colon\, t < i \leq k \text{ and } C^*_{i-1} = A^*_i\}$ denote the set of just-on-time jobs in the block $B[t, k]$. So we have $S^*_t = A^*_t$ and $S^*_j = A^*_j = C^*_{j-1}$ for all $j \in K$. Our next result, Lemma 5.4, implies that the first (job t) and all just-on-time jobs in a block (i.e., elements of K) are either all in J or all outside J.

Lemma 5.4. *If $B[t, k]$ is a block, then either $\{t\} \cup K \subseteq J$ or $\{t\} \cup K \subseteq B[t, k] \backslash J$.*

Proof. Let $j \in K$. We have $C^*_{j-1} = A^*_t + \sum_{i=t}^{j-1} r_i$ since t is on time, and there is no idle time between t and j. We obtain $0 = C^*_{j-1} - A^*_j = A^*_t + \sum_{i=t}^{j-1} r_i - A^*_j$. Since $\sum_{i=t}^{j-1} r_i$ is integer, $A^*_j - A^*_t$ must be integer. This implies that if $j \in J$, then $t \in J$, and if $j \notin J$, then $t \notin J$. □

Lemma 5.5 will be used to prove Lemmas 5.6 and 5.7.

Lemma 5.5. *Let $k \in \{2, \ldots, n+1\}$ be such that $A^*_k \notin \mathbb{Z}$. Then for every realization of the processing durations such that $C^*_{k-1} = A^*_k$, there is an early job $j < k$.*

Proof. Let $\mathbf{r}$ be a realization of $\mathbf{p}$. By contradiction, assume there is no early job before job k. Then $C^*_{k-1} = A^*_1 + \sum_{i=1}^{k-1} r_i = A^*_k$. This implies $A^*_k \in \mathbb{Z}$ (since $A^*_1 = 0$ and $\sum_{i=1}^{k-1} r_i$ are integer), a contradiction. □

In Lemmas 5.6 and 5.7 below we prove that no event change occurs for any just-on-time job. Therefore Lemma 5.8 states that no event change occurs for any job.

Lemma 5.6. *Let $k \in \{2, \ldots, n+1\}$. For every realization of the processing durations such that $C^*_{k-1} = A^*_k$, if there exists an early job $j < k$, then $C'_{k-1} = A'_k$ and $C''_{k-1} = A''_k$.*

Proof. Let $\mathbf{r}$ be a realization of $\mathbf{p}$. Let t be the last early job before k, so $B[t, k]$ is a block. As explained above, let $K = \{i\colon\ t < i \le k \text{ and } C^*_{i-1} = A^*_i\}$ be the set of just-on-time jobs between t and k. By Lemma 5.4, either (i) $\{t\} \cup K \subseteq J$ or (ii) $\{t\} \cup K \subseteq B[t, k] \backslash J$.

(i) $\{t\} \cup K \subseteq J$. First, by induction we show that $C'_j = C^*_j - \Delta$ for all $j \in B[t, k]$. Indeed, $C'_{t-1} \le C^*_{t-1} < A^*_t - 2\Delta < A'_t$ (by Lemmas 5.1 and 5.2) so $S'_t = A'_t = A^*_t - \Delta$ and $C'_t = A^*_t - \Delta + r_t = C^*_t - \Delta$. Consider $t < j \in B[t, k]$. By inductive assumption, $C'_{j-1} = C^*_{j-1} - \Delta$. If $j \in K$, then $j \in J$ and $A'_j = A^*_j - \Delta$, so $S'_j = \max\{C'_{j-1}, A'_j\} = \max\{C^*_{j-1}, A^*_j\} - \Delta = C^*_{j-1} - \Delta$. Otherwise, $j \notin K$, i.e., j is late. Then by Lemma 5.2, $C^*_{j-1} > A^*_j + 2\Delta$. So $A'_j \le A^*_j < C^*_{j-1} - 2\Delta = C'_{j-1} - \Delta$, and hence $S'_j = C'_{j-1} = C^*_{j-1} - \Delta$. In both cases, $C'_j = S'_j + r_j = C^*_{j-1} - \Delta + r_j = \max\{C^*_{j-1}, A^*_j\} + r_j - \Delta = C^*_j - \Delta$, completing our inductive proof. This implies that $C'_{k-1} = C^*_{k-1} - \Delta = A^*_k - \Delta = A'_k$ since $k \in K \subseteq J$ so $C'_{k-1} = A'_k$ as claimed.

Similarly, by induction we show that $C''_j = C^*_j + \Delta$ for all $j \in B[t, k]$. Indeed, $C''_{t-1} \le C^*_{t-1} + \Delta < A^*_t < A^*_t + \Delta = A''_t$ (by Lemmas 5.1 and 5.2), so $S''_t = A''_t = A^*_t + \Delta$ and $C''_t = A^*_t + \Delta + r_t = C^*_t + \Delta$. Consider $t < j \in B[t, k]$. By inductive assumption, $C''_{j-1} = C^*_{j-1} + \Delta$. If $j \in K$, then $j \in J$ and $A''_j = A^*_j + \Delta$, so $S''_j = \max\{C''_{j-1}, A''_j\} = \max\{C^*_{j-1}, A^*_j\} + \Delta = C^*_{j-1} + \Delta$. Otherwise, $j \notin K$, i.e., j is late. Then by Lemma 5.2 $C^*_{j-1} > A^*_j + 2\Delta$. So $A''_j \le A^*_j + \Delta < C^*_{j-1} - \Delta = C''_{j-1} - 2\Delta$, and hence $S''_j = C''_{j-1} = C^*_{j-1} + \Delta$, completing our inductive proof. In both cases, $C''_j = S''_j + r_j = C^*_{j-1} + \Delta + r_j = \max\{C^*_{j-1}, A^*_j\} + r_j + \Delta = C^*_j + \Delta$. This implies that $C''_{k-1} = C^*_{k-1} + \Delta = A^*_k + \Delta = A''_k$ since $k \in K \subseteq J$, so $C''_{k-1} = A''_k$ as claimed.

(ii) $\{t\} \cup K \subseteq B[t, k] \backslash J$. First, by induction we show that $C'_j = C^*_j$ for all $j \in B[t, k]$. Indeed, $C'_{t-1} \le C^*_{t-1} < A^*_t - 2\Delta < A^*_t = A'_t$ (by Lemmas 5.1 and 5.2), so $S'_t = A'_t = A^*_t$ and $C'_t = A^*_t + r_t = C^*_t$. Consider $t < j \in B[t, k]$. By inductive assumption, $C'_{j-1} = C^*_{j-1}$. If $j \in K$, then $j \notin J$ and $A'_j = A^*_j$, so $S'_j = \max\{C'_{j-1}, A'_j\} = \max\{C^*_{j-1} - A^*_j\} = C^*_{j-1}$. Otherwise, $j \notin K$, i.e., j is late. Then by Lemma 5.2, $C^*_{j-1} > A^*_j + 2\Delta$. So $A'_j \le A^*_j < C^*_{j-1} - 2\Delta = C'_{j-1} - 2\Delta$, and hence $S'_j = C'_{j-1} = C^*_{j-1}$. In both cases, $C'_j = S'_j + r_j = C^*_{j-1} + r_j = \max\{C^*_{j-1}, A^*_j\} + r_j = C^*_j$, completing our inductive proof. This implies that $C'_{k-1} = C^*_{k-1} = A^*_k = A'_k$, since $k \in K$ and $k \notin J$, so $C'_{k-1} = A'_k$ as claimed.

Similarly, by induction we show that $C''_j = C^*_j$ for all $j \in B[t, k]$. Indeed, $C''_{t-1} \le C^*_{t-1} + \Delta < A^*_t = A''_t$ (by Lemmas 5.1 and 5.2), so $S''_t = A''_t = A^*_t$ and $C''_t = A^*_t + r_t = C^*_t$. Consider $t < j \in B[t, k]$. By inductive assumption, $C''_{j-1} = C^*_{j-1}$. If $j \in K$, then $j \notin J$ and $A''_j = A^*_j$, so $S''_j = \max\{C''_{j-1}, A''_j\} = \max\{C^*_{j-1}, A^*_j\} = C^*_{j-1}$. Otherwise, $j \notin K$, i.e., j is late. Then by Lemma 5.2, $C^*_{j-1} > A^*_j + 2\Delta$. So $A''_j \le A^*_j + \Delta < C^*_{j-1} - \Delta = C''_{j-1} - \Delta$, and hence $S''_j = C''_{j-1} = C^*_{j-1}$, completing our inductive proof. In both cases, $C''_j = S''_j + r_j = C^*_{j-1} + r_j = \max\{C^*_{j-1}, A^*_j\} + r_j = C^*_j$. This implies that $C''_{k-1} = C^*_{k-1} = A^*_k = A''_k$, since $k \in K$ and $k \notin K$, so $C''_{k-1} = A''_k$ as claimed. □

Lemma 5.7. *Let $k \in \{2, \ldots, n+1\}$. For every realization of the processing durations such that $C^*_{k-1} = A^*_k$, we have $C'_{k-1} = A'_k$ and $C''_{k-1} = A''_k$.*

Proof. If there is an early job before k, then the result follows from Lemma 5.6. Otherwise, $B[1, k]$ is a block. Therefore $C^*_{k-1} = A^*_1 + \sum_{i=1}^{k-1} r_i = A^*_k$. Furthermore, $A^*_k \in \mathbb{Z}$ by Lemma 5.5 so $k \notin J$. Therefore $\{1\} \cup K \subseteq B[1, k] \backslash J$ by Lemma 5.4, and hence the result follows in a similar way to part (ii) of the proof for Lemma 5.6. □

Our next result establishes that no event change occurs for any job and directly follows from Lemmas 5.3 and 5.7. We define the sign of a real number x as $\operatorname{sign}(x) = 1$ if $x > 0$; 0 if $x = 0$; and -1 if $x < 0$.

Lemma 5.8. *For every job $j = 2, \ldots, n+1$, and every realization of the processing durations,*

$$\operatorname{sign}(C'_{j-1} - A'_j) = \operatorname{sign}(C''_{j-1} - A''_j) = \operatorname{sign}(C^*_{j-1} - A^*_j).$$

Lemma 5.9 below gives a consequence on the objective function of this no-event change result.

Lemma 5.9. *F changes linearly with Δ between $\mathbf{A}'$ and $\mathbf{A}''$.*

Proof. There is a no-event change when moving from $\mathbf{A}'$ to $\mathbf{A}''$ by Lemma 5.8. Therefore for every realization $\mathbf{r}$ of the processing duration vector $\mathbf{p}$, $F(\cdot \mid \mathbf{p} = \mathbf{r})$ changes linearly with Δ between $\mathbf{A}'$ and $\mathbf{A}''$. Hence F, $F(\cdot) = \mathrm{E}_p[F(\cdot \mid \mathbf{p})]$, also changes linearly with Δ between $\mathbf{A}'$ and $\mathbf{A}''$. □

Theorem 5.1 (Appointment Vector Integrality). *If the processing durations are integer random variables, then there exists an optimal appointment vector that is integer.*

Proof. By Lemma 4.3 we know that there exists an optimal appointment schedule in the set $\mathscr{K} = \{\mathbf{A} \in \mathbb{R}^{n+1}\colon \underline{\mathbf{A}} \le \mathbf{A} \le \bar{\mathbf{A}}\}$. Let $\mathscr{A}$ denote the set of all such optimal appointment vectors in $\mathscr{K}$, so $\mathscr{A}$ is nonempty, bounded and closed, since by Lemma 4.2, F is continuous. For $\mathbf{A} \in \mathscr{A}$, let

$$I(\mathbf{A}) = \begin{cases} \min\{A_j\colon j \in \{2, \ldots, n+1\} \text{ and } A_j \notin \mathbb{Z}\} & \text{if } \mathbf{A} \notin \mathbb{Z}^{n+1}, \\ n * \bar{p}_{\max} + 1 & \text{if } \mathbf{A} \in \mathbb{Z}^{n+1}. \end{cases}$$

We claim $I(\cdot)$ is upper semicontinuous (usc) on the compact set $\mathscr{A}$. If $\mathbf{A} \in \mathscr{A} \cap \mathbb{Z}^{n+1}$, then $I(\mathbf{A}) = h + 1 \ge I(\mathbf{B})$ for all $\mathbf{B} \in \mathscr{A}$, implying that $I(\cdot)$ is usc at $\mathbf{A}$. Otherwise $\mathbf{A} \in \mathscr{A} \backslash \mathbb{Z}^{n+1}$, and let $I(\mathbf{A}) = A_k$. For any $\epsilon > 0$, let $\delta = \min\{\epsilon, I(\mathbf{A}) - \lfloor A_k \rfloor, \lceil A_k \rceil - I(\mathbf{A})\} > 0$. For all $\mathbf{B} \in \mathscr{A}$, $\|\mathbf{B} - \mathbf{A}\| < \delta$ implies $B_k > A_k - \delta \ge A_k - (I(\mathbf{A}) - \lfloor A_k \rfloor) = \lfloor A_k \rfloor$ and $B_k < A_k + \delta \le A_k + \lceil A_k \rceil - I(\mathbf{A}) = \lceil A_k \rceil$. Therefore B_k is fractional, so $I(\mathbf{B}) \le B_k \le A_k + \epsilon = I(\mathbf{A}) + \epsilon$. Therefore $I(\cdot)$ is usc at $\mathbf{A} \in \mathscr{A} \backslash \mathbb{Z}^{n+1}$. This completes the proof that $I(\cdot)$ is usc on $\mathscr{A}$.

The fact that $I(\cdot)$ is usc and $\mathscr{A}$ is compact implies that there exists an element $\mathbf{A}^*$ of $\mathscr{A}$ maximizing $I(\cdot)$. For contradiction, assume $\mathbf{A}^* \notin \mathbb{Z}^{n+1}$. Let $f = \min\{i\colon A_i^* = I(\mathbf{A}^*)\}$, so for all $j < f$, $A_j^* < I(\mathbf{A}^*)$ and thus $A_j^* \in \mathbb{Z}$. Let $\mathbf{A}'$ and $\mathbf{A}''$ be the schedules derived from $\mathbf{A}^*$ as defined at the beginning of this section. By optimality, $F(\mathbf{A}^*) \le F(\mathbf{A}')$ and $F(\mathbf{A}^*) \le F(\mathbf{A}'')$. But by Lemma 5.9, $F(\mathbf{A}^*)$ changes linearly with Δ between $\mathbf{A}'$ and $\mathbf{A}''$. Hence we must have $F(\mathbf{A}^*) = F(\mathbf{A}') = F(\mathbf{A}'')$. Note that $\mathbf{A}'' \ge \mathbf{A}^* \ge \underline{\mathbf{A}}$ and, for every $j \in J$, $A_j'' = A_j^* + \Delta < \lceil A_j^* \rceil \le \bar{A}_j$ so $\mathbf{A}'' \le \bar{\mathbf{A}}$. This shows that $\mathbf{A}'' \in \mathscr{K}$ and therefore $\mathbf{A}'' \in \mathscr{A}$. But $I(\mathbf{A}^*) = A_f^* < A_f^* + \Delta = A_f'' = I(\mathbf{A}'')$, i.e., $I(\mathbf{A}^*) < I(\mathbf{A}'')$, a contradiction with the definition of $\mathbf{A}^*$. □

Remark 5.1. Linear overage and underage costs are essential for the integrality of an optimal appointment vector. Consider the following example with quadratic costs. Let $n = 1$ and $F(\mathbf{A}) = \mathrm{E}_p[o_1((C_1 - A_2)^+)^2 + u_1((A_2 - C_1)^+)^2]$ with $o_1 = u_1 = 1$; and let $\mathrm{Prob}\{p_1 = 1\} = \mathrm{Prob}\{p_1 = 2\} = \frac{1}{2}$. Then $F(\mathbf{A}) = \mathrm{E}_p[2(C_1 - A_2)^2]$, $C_1 = p_1$, and the optimum is $A_2^* = \mathrm{E}_p(p_1) = \frac{3}{2}$, which is not integer.

6. L-convexity.

We start by investigating an important property of our objective function: submodularity (see, e.g., Fujishige [13], Topkis [35], Murota [22]).

Definition 6.1. A function $f\colon \mathbb{Z}^q \to \mathbb{R}$ is *submodular* iff $f(z) + f(y) \ge f(z \vee y) + f(z \wedge y)$ for all $z, y \in \mathbb{Z}^q$, where $z \vee y = (\max(z_i, y_i)\colon 0 \le i \le q) \in \mathbb{Z}^q$, $z \wedge y = (\min(z_i, y_i)\colon 0 \le i \le q) \in \mathbb{Z}^q$ (Murota [22]).

We now define a property of an appointment vector and a realization of the processing durations that will play an important role in this section.

Definition 6.2. A quadruple (i, j, k, l) is a *submodularity obstacle* for appointment schedule $\mathbf{A}$ and a realization $\mathbf{r}$ of the processing durations if

- $1 \le i < j < k < l \le n + 1$;
- the cost coefficients satisfy $o_{j-1} + u_{j-1} + \sum_{j \le t < k-1} o_t < u_{k-1}$; and, in schedule $\mathbf{A} \mid \mathbf{p} = \mathbf{r}$;
- both jobs i and j are on time;
- job l is the last job that starts on time before job $n + 1$;
- there is no idle time between jobs i and j;
- there is positive idle time between jobs j and l; and
- job k is the first early job after j.

Proposition 6.1. *For any realization* $\mathbf{r}$ *of the processing durations, the function* $F(\cdot \mid \mathbf{p} = \mathbf{r})$ *is submodular if and only if there is no submodularity obstacle for any integer appointment vector* $\mathbf{A}$.

Proof. Let $\mathbf{r}$ be any realization of the processing durations $\mathbf{p}$. By the proof of Theorem 6.19 from Murota [22], $F(\cdot \mid \mathbf{p})$ is submodular if and only if

$$F(\mathbf{A} + \mathbf{1_i} + \mathbf{1_j} \mid \mathbf{p} = \mathbf{r}) - F(\mathbf{A} + \mathbf{1}_i \mid \mathbf{p} = \mathbf{r}) \le F(\mathbf{A} + \mathbf{1}_j \mid \mathbf{p} = \mathbf{r}), -F(\mathbf{A} \mid \mathbf{p} = \mathbf{r}) \tag{2}$$

for each $\mathbf{A} \in \mathbb{Z}^{n+1}$ and let $1 \le i < j \le n + 1$. Let $\mathbf{A} \in \mathbb{Z}^{n+1}$ and $i < j$. Let l be the last on-time job before job $n + 1$. Job l is well defined since job 1 is always on time with $S_1 = A_1$. We consider the following cases for job l.

(a) $1 \le l < j \le n + 1$; i.e., job j is late;
(b) $l = j$; i.e., job j is on time, and all the jobs after job j are late;
(c) $j < l \le n + 1$.

To ease notation we use $(\cdot \mid \mathbf{r})$ to denote schedule $(\cdot \mid \mathbf{p} = \mathbf{r})$. We now verify the submodular inequality (2) in each case.

Case (a) ($l < j \le n+1$). Job j is late for both schedules $(\mathbf{A} \mid \mathbf{r})$ and $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$, and job j remains not early when A_j is replaced with A_j+1; therefore, $F(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r}) = -o_{j-1}$ and $F(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A} \mid \mathbf{r}) = -o_{j-1}$. As a result, (2) holds with equality.

Case (b) ($l = j \le n+1$). Job j is the last on-time job for schedule $(\mathbf{A} \mid \mathbf{r})$, and $(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r})$ pushes every job after job $j-1$ to the right by one unit. Therefore, $F(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A} \mid \mathbf{r}) = u_{j-1} + o_j + o_{j+1} + \cdots + o_n \ge 0$. If there is an idle slot between i and j, then job j will still be on time in schedules $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$ and $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$. Since every job after job $j-1$ in $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$ will also be pushed to the right by one unit, $F(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r}) = u_{j-1} + o_j + o_{j+1} + \cdots + o_n$, and (2) holds with equality. Otherwise, there is no idle slot between i and j. Then job j will be late in schedule $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$ but on time in schedule $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$, and all jobs $k > j$ have the same start times in both schedules $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$ and $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$. Therefore, $F(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r}) = -o_{j-1} \le 0$, and inequality (2) holds.

Case (c) ($j < l \le n+1$). If job j is late in schedule $(\mathbf{A} \mid \mathbf{r})$, then it is also late in schedule $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$, and it remains not early when A_j is replaced with A_j+1. Therefore, $F(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A} \mid \mathbf{r}) = -o_{j-1}$ and $F(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r}) = -o_{j-1}$. As a result, (2) holds with equality. Therefore, assume that job j is on time in schedule $(\mathbf{A} \mid \mathbf{r})$. If there is positive idle time between i and j in schedule $(\mathbf{A} \mid \mathbf{r})$, then j remains on time in schedule $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$; hence it also remains on time in $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$ and $(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r})$, and (2) holds with equality. Therefore, we assume that there is no idle time between i and j. We consider two subcases, CR1 and CR2, for the right-hand side $F(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A} \mid \mathbf{r})$, and three subcases, CL1, CL2, and CL3, for the left-hand side $F(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$, in schedule $(\mathbf{A} \mid \mathbf{r})$:

(CR1) there is no idle time between j and l;
(CR2) there is positive idle time[5] between j and l;
(CL1) job i is on-time;
(CL2) job i is late, and there is no idle time between j and l;
(CL3) job i is late and there is positive idle time between j and l.

In CR1, the time interval $[A_j, A_j+1]$ is idle in schedule $(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r})$ and every job $j, j+1, \ldots, n$ incurs one more unit of overtime in schedule $(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r})$ than in schedule $(\mathbf{A} \mid \mathbf{r})$ since all jobs between j and l are not early and all jobs after l are late. Hence, in CR1, $F(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A} \mid \mathbf{r}) = u_{j-1} + o_j + o_{j+1} + \cdots + o_n$.

In CR2, there is an early job k between j and l. Choose k to be the first early job after j so $j < k < l$. Similarly to CR1, the time interval $[A_j, A_j+1]$ is idle in schedule $(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r})$ and every job $j, j+1, \ldots, k-1$ incurs one more unit of overtime in schedule $(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r})$ than in schedule $(\mathbf{A} \mid \mathbf{r})$ since all jobs between j and k are not early. Furthermore, job $k-1$ incurs one less unit of idle time in schedule $(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r})$ than in schedule $(\mathbf{A} \mid \mathbf{r})$ since k remains not late in schedule $(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r})$. Hence, in CR2, $F(\mathbf{A}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A} \mid \mathbf{r}) = u_{j-1} + o_j + o_{j+1} + \cdots + o_{k-3} + o_{k-2} - u_{k-1}$.

In CL1, job i remains on time in both schedules $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$ and $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$, and because there is no idle time between i and j, job j is late in schedule $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$ but on time in schedule $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$. Therefore, schedule $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$ will have one unit less overtime (just before job j) than schedule $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$. Hence, in CL1, $F(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{p}) - F(\mathbf{A}+\mathbf{1_i} \mid \mathbf{p}) = -o_{j-1}$.

In CL2, job j is just-on-time in schedule $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$ but one time-unit early in schedule $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$. In schedule $(\mathbf{A} \mid \mathbf{r})$, all jobs between j and l are not early (since there is no idle time between j and l), and all jobs after l are late (since l is the last on-time job). Furthermore, all jobs after j are late in schedule $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$ and therefore also late in schedule $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$ because there is no idle time between i and j in schedule $(\mathbf{A} \mid \mathbf{r})$. As a result, schedule $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$ has an idle slot just before A_j+1 and one more unit of overtime for each job $j, \ldots, n+1$ than schedule $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$. Hence, in CL2, $F(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r}) = u_{j-1} + o_j + o_{j+1} + \cdots + o_n$.

Similarly to CL2, in CL3, job j is just-on-time in schedule $(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r})$ but one time-unit early in schedule $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$. Furthermore, there is a first early job k between $j+1$ and l since there is positive idle time between j and l in schedule $(\mathbf{A} \mid \mathbf{r})$. The time interval $[A_j, A_j+1]$ is idle in schedule $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$ and every job $j, j+1, \ldots, k-1$ incurs one more unit of overtime in schedule $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$ than in schedule $(\mathbf{A}+\mathbf{1_i}+ \mid \mathbf{r})$. Furthermore, job $k-1$ incurs one less unit of idle time in schedule $(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r})$ than in schedule $(\mathbf{A}+\mathbf{1_i}+ \mid \mathbf{r})$. Hence, in CL3, $F(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j} \mid \mathbf{r}) - F(\mathbf{A}+\mathbf{1_i} \mid \mathbf{r}) = u_{j-1} + o_j + o_{j+1} + \cdots + o_{k-3} + o_{k-2} - u_{k-1}$. Note that we have the same job k as in CR2.

[5] This means that there is at least one idle slot available between the jobs under consideration. In this case, there exists at least one job which starts on time in the interval.

As a result,

$$F(\mathbf{A}+\mathbf{1_i}+\mathbf{1_j}\mid\mathbf{p})-F(\mathbf{A}+\mathbf{1_i}\mid\mathbf{p})-(F(\mathbf{A}+\mathbf{1_j}\mid\mathbf{p})-F(\mathbf{A}\mid\mathbf{p}))$$
$$=\begin{cases}-o_{j-1}-(u_{j-1}+o_j+o_{j+1}+\cdots+o_n)\le 0 & \text{if CR1 and CL1,}\\ 0 & \text{if CR1 and CL2,}\\ -o_{j-1}-(u_{j-1}+o_j+o_{j+1}+\cdots+o_{k-3}+o_{k-2}-u_{k-1}) & \text{if CR2 and CL1,}\\ 0 & \text{if CR2 and CL3.}\end{cases}$$

If there is no submodularity obstacle, then inequality $-o_{j-1}\le u_{j-1}+o_j+o_{j+1}+\cdots+o_{k-3}+o_{k-2}-u_{k-1}$ in CR2 and CL1 is satisfied, and $F(\cdot\mid\mathbf{p})$ is submodular.

Conversely, if $F(\cdot\mid\mathbf{p})$ is submodular, then $-o_{j-1}\le u_{j-1}+o_j+o_{j+1}+\cdots+o_{k-3}+o_{k-2}-u_{k-1}$ for all jobs $i<j<k<l$ such that j is on time, there is no idle time between i and j, there is positive idle time between j and l, and job i is on-time; i.e., there is no submodularity obstacle for the appointment vector $\mathbf{A}$ and processing duration realization $\mathbf{r}$; hence there cannot be a submodularity obstacle. □

Corollary 6.1. *If there is no submodularity obstacle for any integer appointment vector* $\mathbf{A}$ *and processing duration realization* $\mathbf{r}$, *then* F *is submodular.*

Proof. The result holds since submodularity is preserved under expectation, $F(\cdot)=\mathrm{E}_p[F(\cdot\mid\mathbf{p})]$, and by Proposition 6.1, $F(\cdot\mid\mathbf{p})$ is submodular if there is no submodularity obstacle for any integer appointment vector $\mathbf{A}$ and processing duration realization $\mathbf{r}$. □

A submodularity obstacle is a very specific configuration, and it does not exist with reasonable cost structures such as nonincreasing u_is ($u_{i+1}\le u_i$ for all i) or nonincreasing (o_i+u_i)s ($o_{i+1}+u_{i+1}\le o_i+u_i$ for all i). To capture these cost structures we define the following.

Definition 6.3. The cost coefficients $(\mathbf{u},\mathbf{o})$ are *α-monotone* if there exist real numbers α_i $(1\le i\le n)$ such that $0\le\alpha_i\le o_i$ and $u_i+\alpha_i$ are nonincreasing in i; i.e., $u_i+\alpha_i\ge u_{i+1}+\alpha_{i+1}$ for all $i=1,\ldots,n-1$.

The following lemma establishes a relation between existence of a submodularity obstacle and α-monotonicity.

Proposition 6.2. *If the cost coefficients* $(\mathbf{u},\mathbf{o})$ *are α-monotone, then there is no submodularity obstacle for any integer appointment vector* $\mathbf{A}$ *and processing duration realization* $\mathbf{r}$.

Proof. Assume $(\mathbf{u},\mathbf{o})$ are α-monotone. It suffices to show that for every $j\in\{2,\ldots,n\}$ there exists $t\ge j+1$ such that $o_{j-1}+u_{j-1}+\sum_{r=j}^{t-1}o_r\ge u_{t-1}$ (because with this cost structure no submodularity obstacle can exist for any configuration of jobs with any integer appointment vector $\mathbf{A}$ and processing duration realization $\mathbf{r}$ (Definition 6.2)). For contradiction, suppose $o_{j-1}+u_{j-1}+\sum_{r=j}^{t-1}o_r<u_{t-1}$ for all $t\ge j+1$. Then

$$\alpha_{t-1}+o_{j-1}+u_{j-1}+\sum_{r=j}^{t-1}o_r<u_{t-1}+\alpha_{t-1}\qquad(\text{add }\alpha_{t-1}\text{ to both sides}),$$

$$\alpha_{t-1}+\alpha_{j-1}+u_{j-1}+\sum_{r=j}^{t-1}o_r<u_{t-1}+\alpha_{t-1}\qquad(\text{since }\alpha_{j-1}\le o_{j-1}),$$

$$\alpha_{j-1}+u_{j-1}<u_{t-1}+\alpha_{t-1}\qquad\left(\text{since }\sum_{r=j}^{t-1}o_r+\alpha_{t-1}\ge 0\right),$$

but this is a contradiction to α-monotonicity. Therefore the result follows. □

Theorem 6.1 (Submodularity). *If the cost vectors* $(\mathbf{u},\mathbf{o})$ *are α-monotone, then* F *is submodular.*

Proof. If the cost vectors $(\mathbf{u},\mathbf{o})$ are α-monotone, then by Proposition 6.2 there is no submodularity obstacle for any integer appointment vector $\mathbf{A}$ and processing duration realization $\mathbf{r}$. Hence the result follows from Corollary 6.1. □

Completion times, start times, and tardiness and their expectations are also submodular:

Corollary 6.2. *The tardiness* T_k, *start time* S_k, *completion time* C_k, *and their expected values* $\mathrm{E}_p[T_k]$, $\mathrm{E}_p[S_k]$, *and* $\mathrm{E}_p[C_k]$ *are submodular functions of* $\mathbf{A}$ *for every* $k=1,\ldots,n$.

Proof. Recall that $F(\cdot \mid \mathbf{p}) = \sum_{i=1}^{n}(o_i T_i + u_i E_i)$. Let $1 \le k \le n$, $u_i = 0$ for all i, and $o_i = 1$ if $i = k$ and 0 otherwise. Then $T_k = F(\cdot \mid \mathbf{p})$. Therefore, T_k is submodular whenever $F(\cdot \mid \mathbf{p})$ is. By Proposition 6.1, $F(\cdot \mid \mathbf{p})$ is submodular if there is no submodularity obstacle. But the chosen u_is and o_is are α-monotone so no submodularity obstacle exists by Proposition 6.2. As a result, $F(\cdot \mid \mathbf{p})$ and hence, T_k is submodular. Next we show S_k is submodular. $S_1 = 0$ and $S_k = A_k + \max\{0, C_{k-1} - A_k\} = A_k + T_{k-1}$ $(1 < k \le n)$ by definition. Since A_k is a scalar and T_k is submodular, S_k is also submodular. Similarly, $C_k = S_k + p_k$ $(1 \le k \le n)$ by definition. Since p_k is a scalar and S_k is submodular, C_k is submodular. Finally, the expected values $\mathrm{E}_p[T_k]$, $\mathrm{E}_p[S_k]$, and $\mathrm{E}_p[C_k]$ are submodular since submodularity is preserved under expectation and T_k, S_k, and C_k are submodular. This completes the proof. □

Remark 6.1. The earliness E_k is not a submodular function of $\mathbf{A}$ in general. To see this, let $\mathbf{A} = (0, 3, 5, 6, 9)$, deterministic processing durations $p_1 = 3$, $p_2 = 2$, $p_3 = 2$, $p_4 = 1$. $E_4(\mathbf{A}) = (A_5 - C_4)^+ = (9 - 8)^+ = 1$, and similarly, $E_4(\mathbf{A} + \mathbf{1_1} + \mathbf{1_2}) = 0$, $E_4(\mathbf{A} + \mathbf{1_1}) = 0$, and $E_4(\mathbf{A} + \mathbf{1_2}) = 0$. Therefore, $1 + 0 = E_4(\mathbf{A}) + E_4(\mathbf{A} + \mathbf{1_1} + \mathbf{1_2}) > E_4(\mathbf{A} + \mathbf{1_1}) + E_4(\mathbf{A} + \mathbf{1_2}) = 0 + 0$. Hence, E_4 is not submodular.

The objective function is not only submodular but also L-convex, an important discrete convexity property. Before we show L-convexity results, we give the definition of L-convexity.

Definition 6.4. $f\colon \mathbb{Z}^q \to \mathbb{R} \cup \{\infty\}$ is *L-convex* iff $f(z) + f(y) \ge f(z \vee y) + f(z \wedge y)$ $\forall z, \forall y \in \mathbb{Z}^q$ and $\exists r \in \mathbb{R}$: $f(z + \mathbf{1}) = f(z) + r$ $\forall z \in \mathbb{Z}^q$ (Murota [22]).

Proposition 6.3. *For any realization* $\mathbf{r}$ *of the processing durations, the function* $F(\cdot \mid \mathbf{p} = \mathbf{r})$ *is L-convex if and only if there is no submodularity obstacle for any integer appointment vector* $\mathbf{A}$ *and realization* $\mathbf{r}$.

Proof. Let $\mathbf{r}$ be a realization of the processing durations. If there is no submodularity obstacle for any integer appointment vector $\mathbf{A}$ and realization $\mathbf{r}$, then $F(\cdot \mid \mathbf{p} = \mathbf{r})$ is submodular by Proposition 6.1, the first property in the definition of L-convexity.

Recall that $F(\mathbf{A} \mid \mathbf{p} = \mathbf{r}) = \sum_{i=1}^{n}(o_i T_i + u_i E_i)$, $T_i = (C_i - A_{i+1})^+$, and $E_i = (A_{i+1} - C_i)^+$. Consider $F(\mathbf{A} + \mathbf{1} \mid \mathbf{p} = \mathbf{r}) = \sum_{i=1}^{n}(o_i T_i^1 + u_i E_i^1)$, where $x_i^1 =$ quantity of interest of job i with appointment vector $\mathbf{A} + \mathbf{1}$ for $x \in \{S, C, T, E\}$. Then $S_i^1 = S_i + 1$ and $C_i^1 = C_i + 1$; hence $T_i^1 = T_i$ and $E_i^1 = E_i$. Therefore, $F(\mathbf{A} + \mathbf{1} \mid \mathbf{p} = \mathbf{r}) - F(\mathbf{A} \mid \mathbf{p} = \mathbf{r}) = 0$. This gives us the second property of the L-convexity definition.

Conversely, if $F(\cdot \mid \mathbf{p} = \mathbf{r})$ is L-convex, then $F(\cdot \mid \mathbf{p})$ must be submodular and by Proposition 6.1 there is no submodularity obstacle for any integer appointment vector $\mathbf{A}$. □

Corollary 6.3. *If there is no submodularity obstacle for any integer appointment vector* $\mathbf{A}$ *and realization* $\mathbf{r}$, *then* $F(\cdot)$ *is L-convex.*

Proof. The claim holds since L-convexity is preserved under expectation, $F(\cdot) = E_p[F(\cdot \mid \mathbf{p})]$, and by Proposition 6.3, $F(\cdot \mid \mathbf{p})$ is L-convex if there is no submodularity obstacle for any integer appointment vector $\mathbf{A}$ and realization $\mathbf{r}$. □

Theorem 6.2 (L-Convexity). *If the cost vectors* $(\mathbf{u}, \mathbf{o})$ *are α-monotone, then* $F(\mathbf{A})$ *is L-convex.*

Proof. If the cost coefficients $(\mathbf{u}, \mathbf{o})$ are α-monotone, then by Proposition 6.2 there is no submodularity obstacle for any integer appointment vector $\mathbf{A}$ and processing duration realization $\mathbf{r}$. Therefore, the result follows from Corollary 6.3. □

7. Algorithms.

Using algorithmic results Murota [22, 23], for minimizing L-convex functions, we can minimize the expected cost F in polynomial time, using a polynomial number of expected cost computations and submodular set minimizations.

Assume the input to our problem consists of the number n of jobs, the cost vectors $\mathbf{u}$ and $\mathbf{o}$, the *horizon* h over which F is to be minimized. Assume also that the processing times are integer and that we have an oracle which computes the expected cost $F(\mathbf{A})$ for any given integer appointment vector $\mathbf{A}$.

Theorem 7.1 (Polynomial Time Algorithm 1). *If the cost vectors* $(\mathbf{u}, \mathbf{o})$ *are α-monotone and the processing durations are integer, then there exists an algorithm which minimizes F using polynomial time and a polynomial number of expected cost evaluations.*

Proof. The appointment vector integrality Theorem 5.1 implies that to minimize F we only need to consider integer appointment vectors. If the cost vectors $(\mathbf{u}, \mathbf{o})$ are α-monotone, then F is an L-convex function by the L-convexity Theorem 6.2. Then F can be minimized in $O(\sigma(n)\, EO\, n^2 \log(\lceil h/2n \rceil))$ time by Iwata's steepest descent scaling algorithm (Murota [22, §10.3.2]) where $\sigma(n)$ is the number of function evaluations required to minimize a submodular *set* function over an n-element ground set and EO is the time needed for an expected cost evaluation. □

When the processing durations are independent, the expected cost of an integer appointment vector can be evaluated efficiently. We use recursive equations for the probability distributions of the start time, completion time, tardiness, and earliness of each job and compute F at an integer point $\mathbf{A}$ in $O(n^2\bar{p}_{\max}^2)$ time.

Theorem 7.2. *If the processing durations are stochastically independent and* $\mathbf{A}$ *is an integer appointment vector, then* $F(\mathbf{A})$ *may be computed in* $O(n^2\bar{p}_{\max}^2)$ *time.*

Proof. The first job starts at time zero so $S_1 = A_1 = 0$, and $C_1 = p_1$; i.e., the distribution of C_1 is that of p_1. Next, we look at the start times S_i $(2 \le i \le n)$. We have $S_i = \max(A_i, C_{i-1})$ so for all $k = 0, 1, \dots, n\bar{p}_{\max}$,

$$\operatorname{Prob}\{S_i = k\} = \begin{cases} 0 & \text{if } k < A_i, \\ \operatorname{Prob}\{C_{i-1} \le k\} & \text{if } k = A_i, \\ \operatorname{Prob}\{C_{i-1} = k\} & \text{if } k > A_i. \end{cases} \tag{3}$$

Note that S_i and p_i are independent because S_i is completely determined by $p_1, p_2, \dots, p_{i-1}$ and $A_1, A_2, \dots, A_i$. Since $C_i = S_i + p_i$, by conditioning on p_i and using independence of p_i and S_i, we obtain for all $k = 0, 1, \dots, n\bar{p}_{\max}$,

$$\operatorname{Prob}\{C_i = k\} = \operatorname{Prob}\{S_i = k - p_i\} = \sum_{j=0}^{\bar{p}_i} \operatorname{Prob}\{S_i = k - j\} \operatorname{Prob}\{p_i = j\}, \tag{4}$$

and $\operatorname{Prob}\{C_{i-1} \le k\} = \operatorname{Prob}\{C_{i-1} = k\} + \operatorname{Prob}\{C_{i-1} \le k-1\}$. For each $i-1$, $\operatorname{Prob}\{C_{i-1} \le k\}$ may be computed in $O((i-1)\bar{p}_{\max})$ time. Hence, $\operatorname{Prob}\{C_i = k\}$ can be computed once we have the distribution of S_i. For each job i and value k, computing $\operatorname{Prob}\{S_i = k\}$ by Equation (3) requires a constant number of operations, and computing $\operatorname{Prob}\{C_i = k\}$ by Equation (4) requires $O(\bar{p}_i + 1)$ operations. Therefore the total number of operations needed for computing the whole start time and completion time distributions for job i is $O(n\bar{p}_{\max}\bar{p}_{\max})$. The distribution of T_i and E_i, and their expected values $\mathrm{E}_p T_i$ and $\mathrm{E}_p E_i$ can then be determined in $O(n\bar{p}_{\max})$ time. Therefore, the objective value $F(\mathbf{A})$ is obtained in $O(n^2\bar{p}_{\max}^2)$ time. □

The running time of the algorithm given in Theorem 7.1 depends on how the distributions of the processing durations are given. Under the common assumption of independent durations, the input to the algorithm includes the distribution of each processing duration p_i, which specifies $\bar{p}_i + 1$ probabilities $\operatorname{Prob}\{p_i = x\}$ for $x = 0, 1, \dots, \bar{p}_i$. In this case, F can be minimized in $O(n^9\bar{p}_{\max}^2 \log \bar{p}_{\max})$ time.

Theorem 7.3 (Polynomial Time Algorithm 2). *If the processing durations are independent, integer-valued random variables, and the cost vectors* $(\mathbf{u}, \mathbf{o})$ *are* α*-monotone, then we can minimize* F *in* $O(n^9\bar{p}_{\max}^2 \log \bar{p}_{\max})$ *time.*

Proof. The horizon h can be taken as $n\bar{p}_{\max} \ge \sum_{i=1}^n \bar{p}_i$, so h is polynomially bounded in the input size. Theorem 7.2 shows that $EO = O(h^2)$ when processing durations are independent. Orlin [25, Theorem 4] shows that $\sigma(n) = O(n^5)$. The result follows from the proof of Theorem 7.1. □

8. Objective function with a due date. Suppose that we are given a due date D for the end of processing, instead of letting the model choose a planned makespan A_{n+1}. We assume D is integer and $0 \le D \le \sum_{i=1}^n \bar{p}_i$. Let $\tilde{\mathbf{A}} = (A_1, A_2, \dots, A_n)$. Then our new objective becomes

$$F^D(\tilde{\mathbf{A}}) = \mathrm{E}_p\left[\sum_{j=1}^{n-1}\big(o_j(C_j - A_{j+1})^+ + u_j(A_{j+1} - C_j)^+\big) + o_n(C_n - D)^+ + u_n(D - C_n)^+\right].$$

We immediately observe that $F(\tilde{\mathbf{A}}, D) = F^D(\tilde{\mathbf{A}})$. Furthermore, F^D has many properties such as discrete convexity, optimal vector integrality, and existence of a polynomial time minimization algorithm.

We verify the properties of F^D. Let $\tilde{\mathcal{K}} = \{\tilde{\mathbf{A}} \in \mathbb{R}^n\colon \tilde{A}_1 = 0, \sum_{j<i} \underline{p}_j \le \tilde{A}_i \le \sum_{j<i} \bar{p}_j \text{ for all } i = 2, \dots, n\}$. Since $F(\tilde{\mathbf{A}}, D) = F^D(\tilde{\mathbf{A}})$, by using our previous results on F we obtain the following for F^D.

Corollary 8.1. (i) *Critical path Lemma 4.1 applies to* F^D.

(ii) *Function* F^D *is continuous.*

(iii) *There exists an optimal appointment schedule* $\tilde{\mathbf{A}}^* \in \tilde{\mathcal{K}}$.

(iv) *There exists an optimal appointment schedule* $\tilde{\mathbf{A}}^*$ *satisfying*

$$\underline{p}_i \le \tilde{A}_{i+1}^* - \tilde{A}_i^* \le \sum_{j\le i} \bar{p}_j - \sum_{j<i} \underline{p}_j \quad \textit{for } i = 1, \dots, n-1.$$

(v) *There exists an optimal appointment vector* $\tilde{\mathbf{A}}^* \in \tilde{\mathscr{K}}$ *with components nondecreasing; i.e.,* $\tilde{A}_i^* \leq \tilde{A}_{i+1}^*$ *for all* $i = 1, \ldots, n-1$.

(vi) $F^D(\tilde{\mathbf{A}})$ *may be computed in* $O(n^2 \bar{p}_{\max}^2)$ *time if processing durations are independent and* $\tilde{\mathbf{A}}$ *is integer.*

Proof. (i) Follows directly from critical path Lemma 4.1.

(ii) Continuity is preserved by projection onto a coordinate subspace. Therefore, the result follows from Lemma 4.2.

(iii) The feasible set for F^D is compact since $0 \leq D \leq \sum_{i=1}^n \bar{p}_i$, and the compactness is preserved by projection onto a coordinate subspace. Therefore, the result follows from Lemma 4.3.

(iv) Follows from Lemma 4.4 (by changing $1 \leq i \leq n$ to $1 \leq i \leq n-1$) and the fact that $0 \leq D \leq \sum_{i=1}^n \bar{p}_i$.

(v) Follows from nondecreasing appointment dates Lemma 4.5 (by changing $1 \leq i \leq n$ to $1 \leq i \leq n-1$) and the fact that $0 \leq D \leq \sum_{i=1}^n \bar{p}_i$.

(vi) Since $F(\tilde{\mathbf{A}}, D) = F^D(\tilde{\mathbf{A}})$, we can compute $F^D(\tilde{\mathbf{A}})$ exactly the same way we compute $F(\mathbf{A})$ with $A_{n+1} = D$. Therefore the result follows from Theorem 7.2. □

We next verify that appointment vector integrality also holds for F^D.

Corollary 8.2 (Appointment Vector Integrality). *If the processing durations are integer random variables and the due date is integer, then there exists an optimal appointment vector that is integer.*

Proof. Let $\tilde{\mathbf{A}}^*$ be any noninteger appointment vector and let $\tilde{A}_f^*$ be the first noninteger component of $\tilde{\mathbf{A}}^*$. As before, we define set J, $\varphi(x)$, Δ, $\tilde{\mathbf{A}}' = (\tilde{A}_1', \ldots, \tilde{A}_n')$, and $\tilde{\mathbf{A}}'' = (\tilde{A}_1'', \ldots, \tilde{A}_n'')$. We consider any realization $\mathbf{r}$ of the processing durations. Then, Lemmas 5.1–5.9 follow for F^D (either directly or by taking $A_{n+1}^* = D$).

By Corollary 8.1 we know that there exists an optimal appointment schedule that is in the set $\tilde{\mathscr{K}} = \{\tilde{\mathbf{A}} \in \mathbb{R}^n\colon \tilde{A}_1 = 0, \sum_{j<i} \underline{p}_j \leq \tilde{A}_i \leq \sum_{j<i} \bar{p}_j$ for all $i = 2, \ldots, n\}$. Let $\tilde{\mathscr{A}}$ denote the set of all such optimal appointment vectors in $\tilde{\mathscr{K}}$, so $\tilde{\mathscr{A}}$ is nonempty, bounded, and closed, since by Corollary 8.1, F^D is continuous. For $\tilde{\mathbf{A}} \in \tilde{\mathscr{A}}$, we define $I(\cdot)$ as before but by changing $\mathbf{A}$ to $\tilde{\mathbf{A}}$ and $\mathbb{Z}^{n+1}$ to $\mathbb{Z}^n$. Then, $I(\cdot)$ is usc on $\tilde{\mathscr{A}}$ since upper semicontinuity is preserved by projection onto a coordinate subspace.

The fact that $I(\cdot)$ is usc and $\tilde{\mathscr{A}}$ is compact implies that there exists an element $\tilde{\mathbf{A}}^*$ of $\tilde{\mathscr{A}}$ maximizing $I(\cdot)$. By contradiction, assume $\tilde{\mathbf{A}}^* \notin \mathbb{Z}^n$. Let $f = \min\{i\colon \tilde{A}_i^* = I(\tilde{\mathbf{A}}^*)\}$, so for all $j < f$, $\tilde{A}_j^* < I(\tilde{\mathbf{A}}^*)$ and thus, $\tilde{A}_j^* \in \mathbb{Z}$. Let $\tilde{\mathbf{A}}'$ and $\tilde{\mathbf{A}}''$ be the schedules derived from $\tilde{\mathbf{A}}^*$ as defined at the beginning of §5 and this proof. By optimality, $F(\tilde{\mathbf{A}}^*) \leq F(\tilde{\mathbf{A}}')$ and $F(\tilde{\mathbf{A}}^*) \leq F(\tilde{\mathbf{A}}'')$. But $F(\tilde{\mathbf{A}}^*)$ changes linearly with Δ between $\tilde{\mathbf{A}}'$ and $\tilde{\mathbf{A}}''$ as Lemma 5.9 applies to F^D. Hence we must have $F(\tilde{\mathbf{A}}^*) = F(\tilde{\mathbf{A}}') = F(\tilde{\mathbf{A}}'')$. Note that $\tilde{A}_i'' \geq \tilde{A}_i^* \geq \sum_{k<i} \underline{p}_k$ for all $i = 1, \ldots, n$ and, for every $j \in J$, $\tilde{A}_j'' = \tilde{A}_j^* + \Delta < \lceil \tilde{A}_j^* \rceil \leq \sum_{k<i} \bar{p}_k$ so $\tilde{\mathbf{A}}_i'' \leq \sum_{k<i} \bar{p}_k$ for all $i = 1, \ldots, n$. This shows that $\tilde{\mathbf{A}}'' \in \tilde{\mathscr{K}}$ and therefore $\tilde{\mathbf{A}}'' \in \tilde{\mathscr{A}}$. But $I(\tilde{\mathbf{A}}^*) = \tilde{A}_f^* < \tilde{A}_f^* + \Delta = \tilde{A}_f'' = I(\tilde{\mathbf{A}}'')$; i.e., $I(\tilde{\mathbf{A}}^*) < I(\tilde{\mathbf{A}}'')$, a contradiction with the definition of $\tilde{\mathbf{A}}^*$. □

Remark 8.1. Integrality of D is crucial for an integer optimal appointment vector. Consider the following example.

$$F^D(\tilde{\mathbf{A}}) = E_p\big[o_1(C_1 - \tilde{A}_2)^+ + u_1(\tilde{A}_2 - C_1)^+ + o_2(C_2 - D)^+ + u_2(D - C_2)^+\big]$$

with $o_1 = u_1 = o_2 = u_2 = 1$, $D = \frac{9}{2}$, $p_2 = 3$ (deterministic p_2), and $p_1 = 1$ with probability $\frac{1}{2}$ and $p_1 = 2$ with probability $\frac{1}{2}$. Then, $F^{9/2}(0, 1) = u_2/4 + o_1/2 + o_2/4$, $F^{9/2}(0, 2) = o_2/4 + u_1/2 + o_2/4$, but $F^{9/2}(0, \frac{3}{2}) = u_1/4 + o_1/4 + o_2/4$.

The next result is on the submodularity and discrete convexity of F^D. Before providing the result, we give a definition of $L^\natural$-convexity (Murota [22]), the type of discrete convexity that F^D has.

Definition 8.1. A function $f\colon \mathbb{Z}^q \to \mathbb{R} \cup \{\infty\}$ is $L^\natural$*-convex* if and only if the function satisfies

$$f(z) + f(y) \geq f((z - \beta\mathbf{1}) \vee y) + f(z \wedge (y + \beta\mathbf{1})) \quad \forall z, y \in \mathbb{Z}^q, \ \forall \beta \geq 0 \quad \text{and} \quad \beta \in \mathbb{Z}.$$

There are a few equivalent definitions available for $L^\natural$-convexity (Murota [22]). We used the one called *translation-submodularity* in Definition 8.1. (Note that translation-submodularity becomes submodularity when $\beta = 0$.) Now, we provide the result for F^D.

Corollary 8.3 ($L^\natural$-convexity). *If the cost coefficients* $(\mathbf{u}, \mathbf{o})$ *are* α*-monotone, then* F^D *is* $L^\natural$*-convex and submodular.*

Proof. Assume that the cost coefficients $(\mathbf{u}, \mathbf{o})$ are α-monotone. Then, by L-convexity Theorem 6.2, F is L-convex. Theorem 7.1 of Murota [22] shows that an L-convex function is $L^\natural$-convex. Hence $\forall\, \mathbf{A}, \mathbf{B} \in \mathbb{Z}^{n+1}$, $\forall\, \beta \geq 0$, and $\beta \in \mathbb{Z}$ we have

$$F(\mathbf{A}) + F(\mathbf{B}) \geq F\big((\mathbf{A} - \beta\mathbf{1}) \vee \mathbf{B}\big) + F\big(\mathbf{A} \wedge (\mathbf{B} + \beta\mathbf{1})\big). \tag{5}$$

The inequality (5) particularly holds for $\mathbf{A} = (\tilde{\mathbf{A}}, D)$ and $\mathbf{B} = (\tilde{\mathbf{B}}, D)$, and by using the fact that $F^D(\cdot) = F(\cdot, \mathbf{D})$ we obtain the following inequalities below for $D \in \mathbb{Z}$, $\forall\, \tilde{\mathbf{A}}, \tilde{\mathbf{B}} \in \mathbb{Z}^n$, $\forall\, \beta \ge 0$, and $\beta \in \mathbb{Z}$:

$$F(\tilde{\mathbf{A}}, D) + F(\tilde{\mathbf{B}}, D) \ge F\big(((\tilde{\mathbf{A}}, D) - \beta\mathbf{1}) \vee (\tilde{\mathbf{B}}, D)\big) + F\big((\tilde{\mathbf{A}}, D) \wedge ((\tilde{\mathbf{B}}, D) + \beta\mathbf{1})\big), \tag{6}$$

$$F(\tilde{\mathbf{A}}, D) + F(\tilde{\mathbf{B}}, D) \ge F\big((\tilde{\mathbf{A}} - \beta\mathbf{1}^n, D - \beta) \vee (\tilde{\mathbf{B}}, D)\big) + F\big((\tilde{\mathbf{A}}, D) \wedge (\tilde{\mathbf{B}} + \beta\mathbf{1}^n, D + \beta)\big), \tag{7}$$

$$F(\tilde{\mathbf{A}}, D) + F(\tilde{\mathbf{B}}, D) \ge F\big(\tilde{\mathbf{A}} - \beta\mathbf{1}^n \vee \tilde{\mathbf{B}}, D\big) + F\big(\tilde{\mathbf{A}} \wedge \tilde{\mathbf{B}} + \beta\mathbf{1}^n, D\big), \tag{8}$$

$$F^D(\tilde{\mathbf{A}}) + F^D(\tilde{\mathbf{B}}) \ge F^D\big((\tilde{\mathbf{A}} - \beta\mathbf{1}^n) \vee \tilde{\mathbf{B}}\big) + F^D\big(\tilde{\mathbf{A}} \wedge (\tilde{\mathbf{B}} + \beta\mathbf{1}^n)\big), \tag{9}$$

where $\mathbf{1}^n$ is a vector of 1s in $\mathbb{Z}^n$. (We used notation $\mathbf{1}^n$ to differentiate it from $\mathbf{1}$, which is a vector of 1s in $\mathbb{Z}^{n+1}$.) Inequality (6) follows from (5) and $\mathbf{A} = (\tilde{\mathbf{A}}, D)$ and $\mathbf{B} = (\tilde{\mathbf{B}}, D)$. In (7) we rewrite the terms of (6) by noting $(\tilde{\mathbf{A}}, D) - \beta\mathbf{1} = (\tilde{\mathbf{A}} - \beta\mathbf{1}^n, D - \beta)$ and $(\tilde{\mathbf{B}}, D) + \beta\mathbf{1} = (\tilde{\mathbf{B}} + \beta\mathbf{1}^n, D + \beta)$. Inequality (8) follows from (7) with component-by-component $\vee$ and $\wedge$ operations. Finally, we obtain inequality (9) by using the fact that $F^D(\cdot) = F(\cdot, \mathbf{D})$. Inequality (9) shows that F^D is $L^\natural$-convex and submodular. (Take $\beta = 0$ for submodularity.) □

Similarly to F, we can minimize F^D by using algorithmic results for $L^\natural$-convexity (Murota [22, 23]), with a polynomial number of expected cost computations and submodular set minimizations. As in the case of F, assume the input to our problem consists of the number n of jobs, the cost vectors $\mathbf{u}$ and $\mathbf{o}$, and the horizon h over which F^D is to be minimized. We also assume that the processing times are integer and that we have an oracle that computes the expected cost $F^D(\tilde{\mathbf{A}})$ for any given integer appointment vector $\tilde{\mathbf{A}}$.

Corollary 8.4 (Polynomial Time Algorithm 1). *If the cost vectors* $(\mathbf{u}, \mathbf{o})$ *are* α*-monotone and the processing durations are integer, then there exists an algorithm that minimizes* F^D *using a polynomial time and polynomial number of expected cost evaluations.*

Proof. The appointment vector integrality Corollary 8.2 implies that we only need to consider integer appointment vectors to minimize F^D. If the cost vectors $(\mathbf{u}, \mathbf{o})$ are α-monotone, then F^D is an $L^\natural$-convex function by the $L^\natural$-convexity Corollary 8.3. Then F^D can be minimized in $O(\sigma(n)\, EO\, n^2 \log(\lceil h/2n \rceil))$ time by Iwata's steepest descent scaling algorithm (Murota [22, §10.3.2]). □

As in the case of F, when the processing durations are independent, we can evaluate the expected cost of an integer appointment vector in $O(n^2 \bar{p}_{\max}^2)$ time by Corollary 8.1. In the case of independent processing durations, the input to the algorithm in Corollary 8.4 includes the distribution of each processing duration p_i and we can minimize F^D in $O(n^9 \bar{p}_{\max}^2 \log \bar{p}_{\max})$ time.

Corollary 8.5 (Polynomial Time Algorithm 2). *If the processing durations are independent, integer-valued random variables and the cost vectors* $(\mathbf{u}, \mathbf{o})$ *are* α*-monotone, then we can minimize* F^D *in* $O(n^9 \bar{p}_{\max}^2 \log \bar{p}_{\max})$ *time.*

Proof. The horizon h can be taken as $n\bar{p}_{\max} \ge \sum_{i=1}^n \bar{p}_i$, so h is polynomially bounded in the input size. Corollary 8.1 shows that $EO = O(h^2)$ when processing durations are independent. Orlin [25, Theorem 4] shows that $\sigma(n) = O(n^5)$. The result follows from the proof of Corollary 8.4. □

9. No-shows and emergency jobs.

No-shows and emergency jobs may have important practical applications and implications. For example, no-shows can be quite important in certain outpatient exams such as MRI scans (Hassin and Mendel [16]). Similarly, emergencies, such as emergency surgeries or examinations, can be a huge factor affecting the planned appointment schedules. With some modifications and assumptions, our model can handle no-shows and emergency jobs (that arrive when the processor is busy) in finding an optimal appointment schedule.

We first discuss no-shows. Suppose that there is some probability no-show$_i$ that job i will not show up. If job i does not show up, then its processing duration becomes zero. On the other hand, if it does show up, then its processing duration will be determined by its distribution. Therefore, all we need to do is update the processing duration distribution p_i of job i to take this no-show possibility into account. We can do so by multiplying $(1 - \text{no-show}_i)$ with $\text{Prob}\{p_i = k\}$ for all $k > 0$ and assign $\text{Prob}\{p_i = 0\} = 1 - \sum_{k>0} \text{Prob}\{p_i = k\}$.

Emergency jobs arrive after the processing starts without any appointments, and they may need to be processed as soon as possible. We take a nonpreemptive approach; i.e., we finish processing the current job first. We assume that emergency jobs may arrive only during processing of another job (i.e., planned or emergency). This is a reasonable assumption when either the ratio of total idle time between planned jobs to total processing durations of planned jobs is small or there are not many emergencies. (Our model falls short in taking into account emergency jobs that arrive during idle time. If the number of emergencies is large, then a more effective

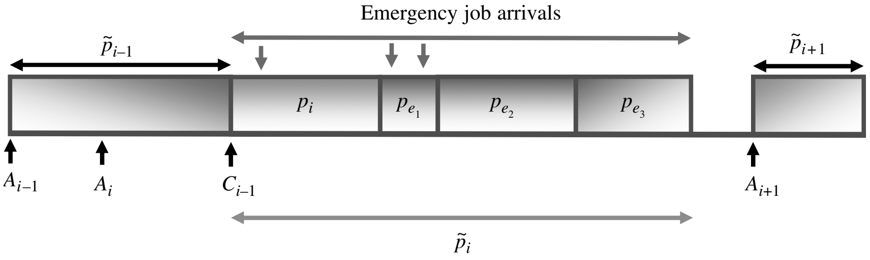

Figure 3. An example schedule with emergency jobs.

approach may be to treat them as a planned job and reserve capacity in advance in the schedule.) Therefore during processing of planned job i, some emergency jobs may arrive, and during these emergencies some other emergency jobs may arrive. All these emergency jobs will be processed back-to-back just after job i and before job $i+1$. Observe that there will be no idle time between the processing of emergency jobs, so we may think of the duration of these emergency jobs processing as a lengthening of job is processing time. Therefore, the problem reduces to finding the new processing duration distribution of job i. Figure 3 shows an example of a schedule with emergency jobs.

We assume that there can be at most a certain number of emergency jobs that can arrive during processing of a job, and the distribution of the number of emergency job arrivals is given by a discrete probability distribution. Furthermore, processing duration distribution of emergency jobs is also given by a discrete probability distribution. There can be at most $m^i_{\max}$ emergency jobs that can arrive during the processing of job i and at most $m^e_{\max}$ emergency jobs that can arrive during the processing of an emergency job. Note that there can be at most $M = m^i_{\max} + m^i_{\max} * m^e_{\max}$ emergency jobs that will be processed after job i and before job $i+1$. Let p_e be the discrete processing duration distribution of emergency jobs. (We use the same processing duration distribution for each emergency job, but one may take p^i_e as the discrete processing duration distribution of emergency jobs arriving during job i.) We denote the total processing duration of emergency jobs that will be processed just after job i as P^i. Then, all we need to do is find the distribution of the new job processing duration $\tilde{p}_i = p_i + P^i$, because once we have $\tilde{p}_i$s, we can minimize $F^D_E(\cdot) = E_{\tilde{p}}[F^D(\cdot \mid \tilde{\mathbf{p}})]$ as we minimize $F^D(\cdot) = E_p[F^D(\cdot \mid \mathbf{p})]$, and $F_E(\cdot) = E_{\tilde{p}}[F(\cdot \mid \tilde{\mathbf{p}})]$ as we minimize $F(\cdot) = E_p[F(\cdot \mid \mathbf{p})]$, and solve the scheduling problem with emergency jobs.

We now obtain the distribution of $\tilde{p}_i = p_i + P^i$. Let m^i $(0 \le m^i \le M)$ be the number of emergency jobs that will be arriving and processed after job i and before job $i+1$. We define $P^i_k = \sum_{j=1}^{k} p^e$ $(1 \le k \le M)$. Distributions P^i_k $(1 \le k \le M)$ can be computed in a recursive manner starting from $P^i_1 = p^e$ and computing $P^i_k = P^i_{k-1} + p^e$ for $1 < k \le M$. Once we have the distribution of P^i_ks we can find the distributions of P^i $(1 \le i \le n)$ as follows:

$$\operatorname{Prob}\{P^i = 0\} = \operatorname{Prob}\{m^i = 0\} + \sum_{k=1}^{M} \operatorname{Prob}\{m^i = k\} \operatorname{Prob}\{P^i_k = 0\},$$

$$\operatorname{Prob}\{P^i = j\} = \sum_{k=1}^{M} \operatorname{Prob}\{m^i = k\} \operatorname{Prob}\{P^i_k = j\} \quad \text{for } j = 1, 2, \ldots, M\bar{p}_{\max}.$$

The last thing we need is to obtain $\tilde{p}_i = p_i + P^i$. This is just a single convolution of random variables p_i (already available) and P^i (just obtained).

10. Current work, future work, and conclusion.

After developing our modeling framework and proving that we can find an optimal appointment schedule in polynomial time, we are now focusing on practical implementation issues. Our objective as a function of continuous appointment vector is nonsmooth but we have shown that it is convex, and we characterized its subdifferential. We have obtained closed-form formulas for the subdifferential as well as for any subgradient. This characterization is very useful; it allows us to develop two very important extensions.

In the first extension (Begen et al. [4]), we relax the perfect information assumption on the probability distributions of processing durations, i.e., we assume that processing duration distributions are not known and can only be statistically estimated on the basis of past data or statistical sampling. Our approach is nonparametric, and we assume no (prior) information about processing duration distributions. We develop a sample-based approach to determine the number of independent samples required to obtain a provably near-optimal solution with high confidence, i.e., the cost of the sample-based optimal schedule is with high probability no more than

$(1+\epsilon)$ times the cost of an optimal schedule determined from knowing the true distributions. This result has important practical implications, as the true processing duration distributions are often not known and only their past realizations or some samples are available.

In the second extension (Begen and Queyranne [2]), we use the subdifferential characterization with independent processing durations and compute a subgradient in polynomial time for any given appointment schedule. This is not a trivial task as the subdifferential formulas include exponentially many terms, and some of the probability computations are complicated. We also obtain an easily computable lower bound on the optimal objective value. Furthermore, we extend computation of the expected total cost (in polynomial time) for any (real-valued) appointment vector. These allow us to use nonsmooth convex optimization techniques to find an optimal schedule. Although we already have a polynomial time algorithm to find an optimal appointment schedule, it is not clear at the moment which technique will work faster in practice. We are also considering hybrid algorithms based on both discrete convexity and nonsmooth convex optimization combined with a special-purpose integer-rounding method. Preliminary versions of these algorithms have been developed. The rounding algorithm takes any fractional solution and rounds it to an integer one with the same or improved objective value. We are planning to implement our algorithms and compare different approaches in computational experiments.

There are many exciting future directions for this research. One is to find an optimal sequence and appointment schedule simultaneously, i.e., given the jobs, determine a sequence and a job appointment schedule minimizing the total expected cost. This problem is likely to be hard, but it may be possible to develop heuristic algorithms with performance guarantees. Studying some special cases for this problem may shed light on the general case. Another objective is to put our findings into practice. We are in contact with local healthcare organizations to apply our results with real data and compare the appointment schedules determined by our methods with current practices.

In this paper, we study a discrete-time version of the appointment scheduling problem and establish discrete convexity properties of the objective function. We prove that the objective function is L-convex under mild assumptions on cost coefficients. Furthermore, we show that there exists an optimal integer appointment schedule minimizing the objective. This result is important as it allows us to optimize only over integer appointment schedules without loss of optimality. All these results on the objective function and optimal appointment schedule enable us to develop a polynomial time algorithm, based on discrete convexity, that, for a given processing sequence, finds an appointment schedule minimizing the total expected cost. When processing durations are stochastically independent we evaluate the expected cost for a given processing order and an integer appointment schedule efficiently in polynomial time. Independent processing durations lead to faster algorithms. Our modeling framework can handle a given due date for the total processing (e.g., end-of-day for an operating room) after which overtime is incurred, instead of letting the model choose an end date. We also extend our model and framework to include no-shows and some emergencies. We believe the framework is portable and applicable to many appointment systems in healthcare as well as in other areas.

Acknowledgments. This research was supported by a Discovery Grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada to the second author and by a PGSD3 NSERC scholarship to the first author. Part of the results in this paper was preliminarily presented in Begen and Queyranne [1]. The authors also thank the area editor and the review team for a constructive review process and their comments that improved the paper.

References

[1] Begen, M. A., M. Queyranne. 2009. Appointment scheduling with discrete random durations. *Proc. 20th Annual ACM—SIAM Sympos. Discrete Algorithms*, SIAM, New York, 845–854.

[2] Begen, M. A., M. Queyranne. 2011. Minimizing a discrete-convex function for appointment scheduling. Working paper, University of Western Ontario, London, Ontario, Canada.

[3] Begen, M. A., M. Queyranne. 2011. Advance multi-period quantity commitment and appointment scheduling. Working paper, University of Western Ontario, London, Ontario, Canada.

[4] Begen, M. A., R. Levi, M. Queyranne. 2011. A sampling-based approach to appointment scheduling. Under review, University of Western Ontario, London, Ontario, Canada.

[5] Begen, M. A., C. T. Ryan, M. Queyranne. 2011. Incentive-based surgery scheduling: Determining optimal number of surgeries. Working paper, University of Western Ontario, London, Ontario, Canada. Forthcoming.

[6] Bendavid, I., B. Golany. 2009. Setting gates for activities in the stochastic project scheduling problem through the cross entropy methodology. *Ann. Oper. Res.* **172**(1) 259–276.

[7] Bosch, P. M. V., D. C. Dietz, J. R. Simeoni. 1999. Scheduling customer arrivals to a stochastic service system. *Naval Res. Logist.* **46**(5) 549–559.

[8] Cardoen, B., E. Demeulemeester, J. Belien. 2010. Operating room planning and scheduling: A literature review. *Eur. J. Oper. Res.* **201**(3) 921–932.

[9] Cayirli, T., E. Veral. 2003. Outpatient scheduling in health care: A review of literature. *Production Oper. Management* **12**(4) 519–549.

[10] Denton, B., D. Gupta. 2003. A sequential bounding approach for optimal appointment scheduling. *IIE Trans.* **35**(11) 1003–1016.

[11] Elhafsi, M. 2002. Optimal leadtime planning in serial production systems with earliness and tardiness costs. *IIE Trans.* **34**(3) 233–243.

[12] Fleischer, L. 2000. Recent progress in submodular function minimization. *OPTIMA: Math. Programming Soc. Newsletter* **64**(9) 1–11.

[13] Fujishige, S. 2005. *Submodular Functions and Optimization*. Elsevier, Amsterdam.

[14] Green, L., S. Savin, B. Wang. 2006. Managing patient service in a diagnostic medical facility. *Oper. Res.* **54**(1) 11–25.

[15] Gupta, D., L. Wang. 2008. Revenue management for a primary-care clinic in the presence of patient choice. *Oper. Res.* **56**(3) 576–592.

[16] Hassin, R., S. Mendel. 2008. Scheduling arrivals to queues: A single-server model with no-shows. *Management Sci.* **54**(3) 565–572.

[17] Iwata, S. 2008. Submodular function minimization. *Math. Programming* **112**(1) 45–64.

[18] Kaandorp, G. C., G. Koole. 2007. Optimal outpatient appointment scheduling. *Health Care Man. Sci.* **10**(3) 217–229.

[19] Luzon, Y., A. Mandelbaum, M. Penn. 2009. Scheduling appointments via fluids control. *Internat. Conf. Model-Based System Engrg.* IEEE, Piscataway, NJ.

[20] McCormick, S. T. 2006. Submodular function minimization. K. Aardal, G. Nemhauser, R. Weismantel, eds. *Handbook on Discrete Optimization*. Elsevier, Amsterdam, 321–391.

[21] Murota, K. 1998. Discrete convex analysis. *Math. Programming* **83**(3) 313–371.

[22] Murota, K. 2003. *Discrete Convex Analysis, SIAM Monographs on Discrete Mathematics and Applications*, Vol. 10. Society for Industrial and Applied Mathematics, Philadelphia.

[23] Murota, K. 2003. On steepest descent algorithms for discrete convex functions. *SIAM J. Optim.* **14**(3) 699–707.

[24] Murota, K. 2009. Recent developments in discrete convex analysis. W. Cook, L. Lovasz, J. Vygen, eds. *Research Trends in Combinatorial Optimization*. Springer-Verlag, Berlin/Heidelberg, 219–260.

[25] Orlin, J. B. 2007. A faster strongly polynomial time algorithm for submodular function minimization. *Math. Programming* **118**(2) 237–251.

[26] Patrick, J., M. L. Puterman, M. Queyranne. 2008. Dynamic multipriority patient scheduling for a diagnostic resource. *Oper. Res.* **56**(6) 1507–1525.

[27] Pinedo, M. 1993. Stochastic scheduling with release dates and due dates. *Oper. Res.* **31**(3) 559–572.

[28] Pinedo, M. 2001. *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, New York.

[29] Robinson, L. W., R. R. Chen. 2003. Scheduling doctors' appointments: Optimal and empirically based heuristic policies. *IIE Trans.* **35** 295–307.

[30] Robinson, L. W., Y. Gerchak, D. Gupta. 1996. Appointment times which minimize waiting and facility idleness. Working paper, DeGroote School of Business, McMaster University, Hamilton, Ontario, Canada.

[31] Sabria, F., C. F. Daganzo. 1989. Approximate expressions for queuing systems with scheduling arrivals and established service order. *Transportation Sci.* **23**(3) 159–165.

[32] Santibanez, P., M. Begen, D. Atkins. 2007. Surgical block scheduling in a system of hospitals: An application to resource and wait list management in a British Columbia health authority. *Health Care Management Sci.* **10**(2) 269–282.

[33] Schutz, H.-J., R. Kolisch. 2008. Capacity allocation for demand of different customer–product combinations with cancellations, no-shows, and overbooking when there is a sequential delivery of service. Working paper, School of Management, Technische Universität, Munchen, Germany.

[34] Strum, D. P., J. H. May, L. G. Vargas. 2000. Modeling the uncertainty of surgical procedure times: Comparison of log-normal and normal models. *Anesthesiology* **92**(4) 1160–1167.

[35] Topkis, D. M. 1978. Minimizing a submodular function on a lattice. *Oper. Res.* **26**(2) 305–321.

[36] Wang, P. P. 1993. Static and dynamic scheduling of customer arrivals to a single-server system. *Naval Res. Logist.* **40**(3) 345–360.

[37] Wang, P. P. 1999. Sequencing and scheduling n customers for a stochastic server. *Eur. J. Oper. Res.* **119**(3) 729–738.

[38] Weiss, E. N. 1990. Models for determining estimated start times and case orderings in hospital operating rooms. *IIE Trans.* **22**(2) 143–150.

[39] Zipkin, P. 2008. On the structure of lost-sales inventory models. *Oper. Res.* **56**(4) 937–944.