Probability in the Engineering and Informational Sciences, **27**, 2013, 25–51.
doi:10.1017/S0269964812000289

THE SEQUENTIAL STOCHASTIC ASSIGNMENT PROBLEM WITH POSTPONEMENT OPTIONS

TIANKE FENG AND JOSEPH C. HARTMAN
Department of Industrial and Systems Engineering
University of Florida, Gainesville, Florida, USA
E-mail: fengtk@ufl.edu; hartman@ise.ufl.edu

The sequential and stochastic assignment problem (SSAP) has wide applications in logistics, finance, and health care management, and has been well studied in the literature. It assumes that jobs with unknown values arrive according to a stochastic process. Upon arrival, a job's value is made known and the decision-maker must immediately decide whether to accept or reject the job and, if accepted, to assign it to a resource for a reward. The objective is to maximize the expected reward from the available resources. The optimal assignment policy has a threshold structure and can be computed in polynomial time. In reality, there exist situations in which the decision-maker may postpone the accept/reject decision. In this research, we study the value of postponing decisions by allowing a decision-maker to hold a number of jobs which may be accepted or rejected later. While maintaining this queue of arrivals significantly complicates the analysis, optimal threshold policies exist under mild assumptions when the resources are homogeneous. We illustrate the benefits of delaying decisions through higher profits and lower risk in both cases of homogeneous and heterogeneous resources.

1. INTRODUCTION AND MOTIVATION

The classic sequential stochastic assignment problem (SSAP) is defined as follows. There are M resources available, having quality values $p_1, p_2, \ldots, p_M$ (or, as a special case, M identical resources all having the same quality value p). A number of jobs, N (typically greater than M), arrive according to a stochastic process and have values $X_1, \ldots, X_N$. These values are known *a priori* to be independently and identically distributed with known distribution function F. Upon the arrival of the ith job at time t, the value of X_i is made known and the decision-maker must immediately accept or reject the job. If a job is accepted and assigned to a resource with quality value p_j, the assignment occurs immediately, yielding a reward $\rho(t)p_jX_i$, where $\rho(t)$ is a discounting function. Declined jobs are lost forever and assigned resources cannot be recalled. The objective is to maximize the expected discounted reward of all assignments.

Although there are many applications where the assignment must occur immediately (or with minimal delay), such as kidney allocation or packet switching, there exist many applications in which the decision-maker may postpone the assignment decision. For example, in the classical secretary problem (often referred to as the labor market analysis or job

© Cambridge University Press 2013 0269-9648/13 $25.00

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

search problem), the offer need not be immediate. Rather, a number of candidates can be interviewed before a decision offer is made. From the perspective of one interviewing for a number of jobs, they may wait for multiple offers before choosing one. In many asset selling or bidding processes, multiple bids may be received before one is accepted. Finally, it may be beneficial in production settings to delay the assignment of a job to a machine given the potential of more lucrative job arrivals. It should be clear that the option to postpone can bring higher rewards and reduce risk. On the other hand, if solutions are time sensitive, postponement may reduce the amount of the reward. The decision-maker must balance these issues.

To our knowledge, no research has been published on the SSAP problem where decisions may be postponed. In this research, we study the value of postponement assuming that jobs arrive according to a Poisson process and we include a discount factor to acknowledge that rewards may be delayed. Under these assumptions, we make the following three contributions to the literature:

1. Model the SSAP with the postponement option for homogeneous resources and prove that the optimal policy is as follows. Suppose that among the jobs that have arrived but have not yet been assigned, the job of greatest value has value x. Then it is optimal to assign this job to a resource iff $x \geq z^{(m,n)}$, where the threshold $z^{(m,n)}$ only depends on m (the number of remaining resources) and n (the number of jobs yet to arrive). Note that the optimal assignment decision does not depend on the precise values of any other jobs that are in the postponement queue.
2. Model the SSAP with the postponement option for heterogeneous resources, investigate the optimal policy, and propose heuristic solutions, based on the insights obtained for the case with homogeneous resources.
3. Illustrate the benefit of the postponement option with both homogeneous and heterogeneous resources.

The paper is organized as follows. After a brief review of the relevant literature in Section 2, we introduce notation, assumptions, and our model in Section 3. In Section 4, the optimal threshold policy for the SSAP with homogeneous resources is developed through induction. We compare this policy with the optimal policy requiring an immediate decision. Conditions under which postponement is superior are identified. In Section 5, the SSAP with heterogeneous resources is discussed and numerical examples illustrate the benefits of postponement.

2. LITERATURE REVIEW

In their seminal work on the SSAP, Derman, Lieberman, & Ross [6] study the case of $N = M$ without a deadline or discounting and define an optimal threshold policy. Specifically, at the initial stage with resource values ordered as $p_1 \leq \cdots \leq p_M$, the support of F can be partitioned into M non-overlapping intervals and the optimal decision is to assign the job valued at x to the resource valued at p_i if x falls within the ith interval. Moreover, the threshold values used to obtain these intervals only depend on M and F and can be computed in time $O(M^2)$. Assignment decisions for later arrivals follow the same rule with a reduced M. The authors show that the threshold structure holds when the assignment reward can be generalized as $f(x, p)$. Albright [2] extends the work to infinite job arrivals and discounting.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

Sakaguchi [24] extends the work of Albright [2] to the case in which jobs are finite in number and arrive according to a non-homogeneous Poisson process. Sakaguchi [25] permits the number of available resources to be unknown. Albright [3] studies the "secretary problem" where the best M secretaries are to be selected from N available secretaries. The model in this work assumes the value of the secretaries come from different distributions, with two successive secretaries' values governed by a Markov Chain. Nakai [15,16] studies the case with the distribution of resources varying according to a partially observable Markov process. Albright [4] considers the case when the parameters of the distribution of the job value are not fully known and allows the parameters to be updated through a Bayesian model. Kennedy [12] permits job values to be dependent. Righter [20] studies the case where each person has independent deadlines and compares this model with the discount model. Righter [21] permits the arrival rate, the job's value, and the variability of job values to change according to independent Markov processes. Additionally, Albright and Derman [1] and Saario [23] studied the asymptotic results of the SSAP.

The dynamic and stochastic knapsack problem (DSKP) with homogeneous sized items studied by Kleywegt and Papastavrou [13] is closely related to SSAP. By setting the costs appropriately, the threshold policy developed there holds for the SSAP with homogeneous resources. Nikolaev and Jacobson [17] extended the results of the SSAP and the DSKP to the case with a random number of arrivals.

A variety of applications utilize the SSAP. For example, the resource can be replaced by houses to sell and jobs can be replaced by purchase offers with random values arriving at random times. Elfving [7] uses this interpretation for the case with $M = 1$. McLay, Jacobson, & Nikolaev [14] applies the optimal policy of the SSAP to the allocation of screening resources to passengers. The studies of the SSAP also bring insight to the organ allocation problem in health care management, including Zenios, Chertow, & Wein [30] and Su and Zenios [26].

The value of postponing decisions is widely identified in the finance (Trigeorgis [28]) and operations management (Van Hoek [29]) literature. In finance, this is generally in the context of real options. As stated by Triantis [27], real options are "opportunities to delay and adjust investments and operating decisions over time in response to resolution of uncertainty." Amram and Kulatilaka [5] and Trigeorgis [28] illustrate the benefits of postponement in reducing uncertainty and increasing profit. although similar, the postponement period in these situations is generally over a much longer time frame (months, quarters, or years) than in our envisioned applications.

3. ASSUMPTIONS AND MODEL DEFINITIONS

With the postponement option, upon the arrival of a job, the decision-maker has three choices upon the arrival of a job: (1) reject the job; (2) accept the job and assign it to a resource; (3) postpone the accept/reject decision. If the decision is to postpone, (1) and (2) are available in the future. The problem ends when either all jobs have arrived or all resources have been assigned.

In this paper, we study the SSAP with the postponement option under the following assumptions:

1. The values $X_1, \ldots, X_N$ are distributed as i.i.d. random variables taking values in $[0, 1]$, with a distribution function F (this information is known *a priori*).
2. Jobs arrive according to a Poisson process with arrival rate λ and there is no deadline;

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

3. At most, M resources can be assigned and a total of N jobs arrive.
4. Rewards are discounted continuously by a positive factor γ, that is, $\rho(t) = \exp(-\gamma t)$.
5. The inter-arrival time of a job is independent of its value.
6. All decisions are implemented instantaneously.

Note that, the case of $\gamma = 0$ reduces to the static problem, as one could wait for all jobs (postpone decisions) and then make assignments at no penalty. We do not consider this case.

We model the problem as a Markov decision process (MDP). With the postponement option, a queue of jobs is maintained in addition to the queue of available resources (as in the traditional SSAP with immediate decisions). Clearly, this queue of jobs greatly complicates the analysis, as it significantly enlarges the decision space.

3.1. State Variable S

Given the number of remaining resources, m, and the number of jobs yet to arrive, n, at time t, the *state* of the system is defined as $S = (\mathbf{x}_m, \mathbf{p}_m, n, t)$. Specifically, $\mathbf{x}_m \in \mathbb{R}^m$ represents the values of the jobs that have arrived but have not been accepted or rejected; similarly, $\mathbf{p}_m \in \mathbb{R}^m$ represents the quality values of resources that have not been allocated. Essentially, $\mathbf{x}_m$ defines a queue of length m whereupon job arrivals are inserted according to their value x such that $\mathbf{x}_m = (x_1, \ldots, x_m)$ and $x_1 \leq x_2 \leq \cdots \leq x_m$; $\mathbf{p}_m$ defines a queue of available resources with $\mathbf{p}_m = (p_1, \ldots, p_m)$ and $p_1 \leq p_2 \leq \cdots \leq p_m$. Let us henceforth refer to a queued job that has the ith smallest value as 'job i' and an available resource that has the jth smallest quality value as 'resource j'. A value of $x_i = 0$ signifies a job with value 0 or the case when there are less than m jobs in the queue. At $t = 0$, $m \equiv M$, $n \equiv N$ and $x_i \equiv 0$, for all i. Denote $\mathbb{S}_m$ as the state space given m available resources.

3.2. Policies, State Transitions, and Decision Epochs

Upon the arrival of a job with value y, the job with the lowest value (x_1 or y) is rejected (and removed from the queue, $\mathbf{x}_m$) as it should be clear that it cannot appear in any optimal solution with m resources. If fewer than m jobs are in the queue, the deletion of $x_1 = 0$ merely entails removing a placeholder. Furthermore, with the delay option, it should be clear that it cannot be optimal to reject any job upon its arrival if its value is greater than x_1, as one can retain it in the queue and delete x_1. We define the two operations of insertion and deletion as *update* and denote it with the operator U, where

$$U(\mathbf{x}_m, y) = \begin{cases} (x_1, x_2, \ldots, x_m) & \text{if} y \leq x_1, \\ (x_2, \ldots, x_i, y, x_{i+1}, \ldots, x_m) & \text{if} x_i \leq y \leq x_{i+1}, i = 1, 2, \ldots, m, \\ (x_2, \ldots, x_m, y) & \text{if} x_m < y. \end{cases} \quad (1)$$

Suppose the state is $(\mathbf{x}_m, \mathbf{p}_m, n, t)$ after updating. The decision-maker can either immediately accept job i and assign it to resource j at time t, receiving a reward, or postpone the decision by a certain amount of time. In the case of immediate assignment, the number of components in $\mathbf{x}_m$ and $\mathbf{p}_m$ decreases by one, respectively. In the case of postponement, there is no immediate change to $\mathbf{x}_m$ or $\mathbf{p}_m$.

Since this is an infinite horizon problem, we can restrict analysis to *stationary deterministic policies*, according to the standard results (see Theorem 6.2.12 of Puterman [18]). Thus, time t can be removed from the state variable S. In this case, the decision-maker

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

should either immediately accept a job and assign it to a resource or postpone until the arrival of a new job. Formally, define the action set for $\mathbb{S}_m$ as:

$$\Lambda(\mathbb{S}_m) = \Big\{\text{accept and assign job } i \text{ to resource } j \text{ with } 1 \le i, j \le m;$$
$$\text{postpone until the arrival of the next job}\Big\}.$$

Define the *decision rule* as $D : \mathbb{S}_m \mapsto \Lambda(\mathbb{S}_m)$. Note that embedded in D is an *assignment rule* $A : \mathbb{S}_m \mapsto \mathbb{S}_{m-1}$ determining which job should be accepted and to which resource the accepted job should be assigned. Decision rule D can be alternatively interpreted as determining whether to apply A immediately or postpone. Define *policy* π as the sequence of decision rules for each decision.

If the decision-maker decides to immediately accept and assign a job, the system transitions to $A(\mathbf{x}_m, \mathbf{p}_m, n)$. If the decision-maker decides to postpone until the arrival of the next job, the system $(\mathbf{x}_m, \mathbf{p}_m, n)$ transitions to $(U(\mathbf{x}_m, y), \mathbf{p}_m, n-1)$. In the former case, m is reduced by 1, and in the latter case, n is reduced by 1. The decision process terminates when either m or n goes to 0. If $m = 0$ and $n \ge 0$, the remaining jobs that have not yet arrived are ignored. If $m > 0$ and $n = 0$, the decision to postpone is no longer beneficial as all jobs have arrived.

Decision epochs (when a decision is made) occur whenever the state changes either at the arrival of a new job or when a job is assigned to a resource. As a result, jobs may be consecutively accepted and assigned to resources at the same point in "time", each belonging to different decision epochs, since the first assignment changes the state and poses a new decision problem to the decision-maker. Moreover, after postponing and receiving a new arriving job, the decision-maker can choose to postpone again.

3.3. Expected Profit and Dynamic Programming Equation

Define Π_{P} as the set of policies with the postponement option. We show below that an optimal policy exists. The optimal policy is not unique, but we denote one optimal policy as π^*. For the sake of convenience, we define the *expected profit* of state $(\mathbf{x}_m, \mathbf{p}_m, n)$ as $V_{m,n}(\mathbf{x}_m, \mathbf{p}_m)$. The *dynamic programming equation* selects the maximum between the expected profit of postponing until the arrival of a new job and the expected profit of immediately accepting a job from the queue and assigning it to a resource:

$$V_{m,n}(\mathbf{x}_m, \mathbf{p}_m) = \max\Big\{\theta \int_0^1 V_{m,n-1}\big(U(\mathbf{x}_m, y), \mathbf{p}_m\big) dF(y), \tag{2}$$
$$r^{A^*}(\mathbf{x}_m, \mathbf{p}_m) + V_{m-1,n}\big(A^*(\mathbf{x}_m, \mathbf{p}_m)\big)\Big\},$$

where $\theta = \lambda/(\gamma + \lambda)$, A^* is the optimal assignment rule, and r^{A^*} is the reward when applying the optimal assignment rule. The boundary conditions are given as:

$$V_{0,n}(\mathbf{x}_m, \mathbf{0}) = 0, \tag{3}$$

$$V_{m,0}(\mathbf{x}_m, \mathbf{p}_m) = \sum_{i=1}^{m} x_i p_i. \tag{4}$$

Equation (3) corresponds to the situation when resources are depleted and (4) corresponds to the situation when no jobs will arrive. Equation (4) is a consequence of the rearrangement inequality (see Theorem 368, Hardy, Littlewood, & Pólya [8]), which states that it is optimal to assign the highest valued jobs to the highest valued resources, respectively.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

4. HOMOGENEOUS RESOURCES

With homogeneous resources, we assume that $p_1 = \cdots = p_m = 1$. Also, the assignment rule A only specifies the queued jobs to be accepted. So in this section, we use 'accept' to mean 'accept and assign'. Furthermore, we can remove $\mathbf{p}_m$ from S. We establish the expected profit (value function) for the SSAP with the postponement option in Section 4.1. Then, we compare it to that of the SSAP without the postponement option in Section 4.2. Finally, we illustrate the difference between the two policies through numerical experiments in Section 4.3.

4.1. SSAP with the Postponement Option

The optimal threshold policy for the SSAP with the postponement option is established through induction. We first establish the result for $m = 1$ and then extend the case to $m \geq 2$. Also, throughout this paper, f' is to be understood at taking the right-hand derivative of a right-hand differentiable function f.

4.1.1. Case 1: $m = 1$. Given $\mathbf{x}_1 = (x_1)$, let $g_{1,n}(x_1)$ (which equals to $V_{1,n}(\mathbf{x}_1)$) be the maximal expected discounted profit that can be obtained when there is one resource remaining, a job of value x_1 queued, and there are n further jobs yet to arrive. The dynamic programming equation is clearly

$$g_{1,n}(x) = \max\Big\{\theta E\Big[g_{1,n-1}\big(\max(x, X)\big)\Big], x\Big\}, \tag{5}$$

with boundary condition $g_{1,0}(x) = x$. The two terms within the curly brackets are the same as those in (2). It is easy to prove that $g_{1,n}(x)$ converges to some $g_1(x)$ as $n \to \infty$, where

$$g_1(x) = \max\Big\{\theta E\Big[g_1\big(\max(x, X)\big)\Big], x\Big\}.$$

Thus, $g_1(x_1)$ is the maximum expected profit that can be obtained when one resource remains, there are an unlimited number of jobs to arrive and one job of value x_1 is queued. Define

$$z^{(1)} = x : x = \theta E[\max(x, X)]. \tag{6}$$

The root is easily seen to exist and be unique. It is straightforward to show by induction on n that $g_{1,n}(x) = x, x \geq z^{(1)}$. Lemma 4.1 presents the characteristics of $g_{1,n}(x)$ that are utilized in the proof of the optimal policy in Theorem 4.1.

LEMMA 4.1: *Let $z^{(1)}$ be the value of x such that $x = \theta\mathbb{E}\big[\max\{x, X\}\big]$. The function sequence defined in (5) possesses the following properties:*

1. *$g_{1,n}(x)$ is strictly increasing in x for $x \in [0, z^{(1)})$ and $n = 1, 2, \ldots$.*
2. *$g_{1,n}(x)$ is strictly increasing in n for $x \in [0, z^{(1)})$.*
3. *$g_{1,n}(x) > x$ for $x \in [0, z^{(1)})$ and $g_{1,n}(z^{(1)}) = z^{(1)}$ for $n = 1, 2, \ldots$.*
4. *$1 > g'_{1,n}(x) > g'_{1,n+1}(x)$ for $x \in (0, z^{(1)})$ and $n = 1, 2, \ldots$.*
5. *$g_{1,n}(x) \to z^{(1)}$ as $n \to \infty$ for $x \in [0, z^{(1)})$.*
6. *$g'_{1,n}(x)$ is non-decreasing in x for $x \in (0, 1)$ and $n = 1, 2, \ldots$.*

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

PROOF: See Appendix A. ■

We now use Lemma 4.1 to prove Theorem 4.1, which states that the optimal policy with $m = 1$ possesses a threshold structure, that is, it is optimal to accept the first arriving job that has a value of at least $z^{(1)}$. If no such job arrives, the job of greatest value, which will have been held in the queue as a postponed job or is the last job to arrive, is accepted.

THEOREM 4.1: *Given $\boldsymbol{x}_1 = x_1$, $m = 1$, and $n = 1, 2, \ldots$, an optimal decision immediately accepts job 1 if $x_1 \geq z^{(1)}$ or otherwise to postpone until the arrival of the next job. The expected profit is $V_{1,n}(x_1) = g_{1,n}(x_1)$. The optimal decision remains the same as $n \to \infty$, with the expected profit as $V_1(x_1) = g_1(x_1)$.*

PROOF: Define the stopping set:

$$\chi = \Big\{x : x \geq \theta \mathbb{E}\Big[\max(x, X)\Big]\Big\}.$$

From Lemma 4.1, it is clear that $\chi = \{x : x \geq z^{(1)}\}$. The stopping set is closed, since $x \in \chi \Rightarrow \max\{x, X\} \in \chi$. From standard results (see Ross [22]), a one-step-look-ahead (OSLA) rule is optimal for all horizons (finite or infinite), i.e., we should accept the first job whose value is at least $z^{(1)}$. It is easy to verify that $V_{1,n}(x_1) = g_{1,n}(x_1)$. As $n \to \infty$, the decision rule remains the same. The convergence of $V_{1,n}(x_1)$ follows from Lemma 4.1. ■

4.1.2. Case 2: $m \geq 2$. In this case, multiple jobs may be retained through postponement. As will be seen, the decision-maker only needs to consider job m (a job of greatest value). We begin with a small lemma for subsequent case.

LEMMA 4.2: *Suppose resources are homogeneous, with $\mathbf{p}_m = (1, \ldots, 1)$. Then, the optimal assignment rule is $A^*(\mathbf{x}_m, \mathbf{p}_m, n) = x_m$. That is, one should always accept the job of greatest value.*

PROOF: See Appendix B. ■

As noted in Section 1, the decision-maker only needs to consider job m (the queued job of greatest value). This conclusion is straightforward for the case with $m > n$, because job m will invariably be assigned, even if all n jobs that are yet to arrive have values greater than that of job m. Knowing this, one may as well assign job m immediately.

The case with $m \leq n$ is much more complicated. Let $V^{\mathrm{POST}}_{m,n}(\mathbf{x}_m)$ denote the supremum over all policies of the expected profit that can be obtained if we start in state $(\mathbf{x}_m, n)$ and initially postpone (i.e., wait for the next arrival before any further assignment of jobs). Define

$$G_{m,n}(\mathbf{x}_m) = V^{\mathrm{POST}}_{m,n}(\mathbf{x}_m) - \Big(x_m + V_{m-1,n}(\mathbf{x}_m)\Big), \tag{7}$$

where $\mathbf{x}_m = (\mathbf{x}_{m-1}, x_m)$. This is indeed the difference of the two expressions in the dynamic programming equation (2) and is non-negative if and only if the postponement option is best.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

We will make much use of the following reduction. We rewrite (7) as

$$G_{m,n}(\mathbf{x}_m) = V^{\text{POST}}_{m,n}(\mathbf{x}_m) - V^{\text{POST}}_{m-1,n}(\mathbf{x}_{m-1}) - x_m + \Big\{V^{\text{POST}}_{m-1,n}(\mathbf{x}_{m-1}) - V_{m-1,n}(\mathbf{x}_{m-1})\Big\}. \tag{8}$$

We show by induction on n, in Theorem (4.2) below, that $V_{m,n}(\mathbf{x}_m)$ can be written in the separable form

$$V_{m,n}(\mathbf{x}_m) = g_{m,n}(x_m) + V_{m-1,n}(\mathbf{x}_{m-1}) = \sum_{k=1}^{m} g_{k,n}(x_k), \tag{9}$$

where $g_{k,n}(x_k), k \geq 2$ is defined as

$$g_{m,n}(x_m) = \begin{cases} \theta g_{m,n-1}(x_m)F(x_m) & \\ +\theta \displaystyle\int_{x_m}^{1} \Big[g_{m,n-1}(y) + g_{m-1,n-1}(x_m) - g_{m-1,n-1}(y)\Big] dF(y), & x_m < z^{(m,n)} \\ x_m, & x_m \geq z^{(m,n)} \end{cases} \tag{10}$$

Suppose (by an induction hypothesis) we know that (9) holds when n is replaced by any of $1, \ldots, n-1$. We rewrite (8) as

$$\begin{aligned} G_{m,n}(\mathbf{x}_m) &= \theta \int_0^1 \Big[V_{m,n-1}\big(U(\mathbf{x}_m, y)\big) - V_{m-1,n-1}\big(U(\mathbf{x}_{m-1}, y)\big)\Big] dF(y) \\ &\qquad - x_m + \Big\{V^{\text{POST}}_{m-1,n}(\mathbf{x}_{m-1}) - V_{m-1,n}(\mathbf{x}_{m-1})\Big\} \\ &= \theta g_{m,n-1}(x_m)F(x_m) \\ &\qquad + \theta \int_{x_m}^1 \Big[g_{m,n-1}(y)dF(y) + g_{m-1,n-1}(x_m) - g_{m-1,n-1}(y)\Big] dF(y) \\ &\qquad - x_m + \Big\{V^{\text{POST}}_{m-1,n}(\mathbf{x}_{m-1}) - V_{m-1,n}(\mathbf{x}_{m-1})\Big\}. \end{aligned} \tag{11}$$

Note that, the term in curly brackets in (11) is always non-positive. In fact, it is either equal to 0 or $G_{m-1,n}(\mathbf{x}_{m-1})$.

At this point, we provide some motivation. Rigor will be added in the proof of Theorem 4.2 below. Consider state $(\mathbf{x}_{m-1}, n-1)$ and suppose that it is optimal to make an assignment. Then it is plausible that it is also optimal to make an assignment in state $(\mathbf{x}_m, n-1)$ and also in states for which x_m is greater. (This plausible statement will be proved in Theorem 4.2.) Assuming this is true, we see that for all $y \geq x_m, g_{m,n-1}(y) = g_{m-1,n-1}(y) = y$. It follows from (11) that $G_{m,n}(\mathbf{x}_m) \leq -(1-\theta)x_m < 0$. So, it is optimal to make an assignment in state $(\mathbf{x}_m, n)$.

The converse of the above conclusion is that if it is optimal to postpone in state $(\mathbf{x}_m, n)$ then it must also be optimal to do so in state $(\mathbf{x}_{m-1}, n-1)$. It is plausible that it should then also be optimal to postpone in state $(\mathbf{x}_{m-1}, n)$. Assuming so, the term in curly brackets is 0 and we see that in the region $G_{m,n}(\mathbf{x}_m) \geq 0$, the value of $G_{m,n}(\mathbf{x}_m)$ is completely determined by x_m alone. Hence, there is a unique value of x_m that makes $G_{m,n}(\mathbf{x}_{m-1}, x_m) = 0$ and this

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

value does not depend on $\mathbf{x}_{m-1}$. We call this root $z^{(m,n)}$. And by the converse statement at the beginning of this paragraph, we have shown that $z^{(m,n)} \leq z^{(m-1,n-1)}$. (By the way, it is interesting that:

$$z^{(m,n)} = V_{m,n}^{\mathrm{POST}}(\mathbf{x}_m) - V_{m-1,n}(\mathbf{x}_{m-1}).$$

This is the cost incurred when job m moves from the queue and is assigned to a resource.)

Before proceeding further with arguments that make the above rigorous, it is helpful to state two straightforward lemmas. The following lemma holds for the problem with heterogeneous resources. Throughout the remainder of the paper we use $\partial/\partial x_i$ to denote a right-hand derivative. This is a necessary technicality because $V_{m,n}(\mathbf{x}_m)$ is not everywhere differentiable. However, its right-hand derivative does exist, and this is what we mean by $\partial V_{m,n}(\mathbf{x}_m, \mathbf{p}_m)/\partial x_i$.

LEMMA 4.3: *The optimal value function $V_{m,n}(\mathbf{x}_m, \mathbf{p}_m)$ is an increasing and convex function of $\mathbf{x}_m$. Moreover the right-hand derivative, $\partial V_{m,n}(\mathbf{x}_m, \mathbf{p}_m)/\partial x_i$, is no greater than p_m. In the homogeneous case of $\mathbf{p}_m = (1, \ldots, 1)$, this derivative is no greater than 1 and can be interpreted as the probability that job i is assigned by an optimal policy.*

Remark: The interpretation of the right-hand derivative as the probability that job i is assigned is ambiguous if the optimal policy is not unique. So let us focus on that optimal policy which would take the same action if x_i were made infinitesimally greater. That is consistent with assigning a job if $x_m = z^{(m,n)}$.

PROOF: See Appendix C. ■

LEMMA 4.4: *$G_{m,n}(\mathbf{x}_m)$ has the following properties:*

(1) *$G_{m,n}(\mathbf{x}_m)$ is a convex and strictly decreasing function of x_m.*
(2) *$G_{m,n}(\mathbf{x}_m) > 0$ when $\mathbf{x}_m = (0, \ldots, 0)$.*
(3) *$G_{m,n}(\mathbf{x}_m) < 0$ when $x_m = 1$.*
(4) *There is a unique x_m such that $G_{m,n}(\mathbf{x}_m) = 0$.*

PROOF: (1) is indicated by (7) and Lemma 4.3. Statements (2) and (3) are trivial, and together with (1) they imply (4). Note that in claiming (4) we are not saying that the root is independent of $\mathbf{x}_{m-1}$. That fact is proved later. ■

Now, we state and prove properties of the optimal policy and the expected profit. We prove the theorem by induction on the value of $m + n$. In proving an induction step for statement (x), we suppose that all statements are true for smaller values of $m + n$, and that statements prior to (x) in the list have been proved for a current value of $m + n$. Define $z^{(m,n)} = 0$ for all $m > n$. This is consistent with the obvious fact that if $m > n$ it is optimal to immediately assign job m.

THEOREM 4.2: *For all m, n, the following are true:*

(a) For $n \geq m$, there exists a unique $x_m = z^{(m,n)}$ such that $G_{m,n}(\mathbf{x}_{m-1}, x_m) = 0$ for all $\mathbf{x}_{m-1}$.
(b) There exists an optimal policy which in state $(\mathbf{x}_m, n)$ makes an assignment (of job m) if and only if $m > n$, or $n \geq m$ and $x_m \geq z^{(m,n)}$.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

(c) $V_{m,n}(\mathbf{x}_m) = g_{m,n}(x_m) + V_{m-1,n}(\mathbf{x}_{m-1})$, *where* $g_{m.n}(x)$ *is defined in* (10).
(d) $z^{(1,1)} = \ldots = z^{(1,n)} = z^{(1)} < 1$;
$z^{(m,n-1)} < z^{(m,n)} < z^{(m-1,n-1)}$, *for all* $n \geq m > 1$.
(e) $1 \geq g'_{m,n}(x) \geq g'_{m-1,n}(x)$, *for all* x.
(f) $z^{(m,n)} \leq z^{(m-1,n)}$.
(g) $g'_{m,n}(x) \leq g'_{m,n-1}(x)$, *for all* x, *with strict inequality if* $x < z^{(m,n)}$.

Remark 1: The statements in the theorem are carefully ordered. Note that, (f) is actually redundant because it is implied by (d) once n increases to $n+1$ in (d). However, in proving (d) we will wish to use (f).

Remark 2: One might wish to present the results for an infinite number of arrivals (from this point we informally call it as "the case of infinite n") as a separate theorem, rather than as a point (h). This result is different because it is not proved by the same induction, and is thus written as a corollary of Theorem 4.2.

PROOF: The proof is by induction on the value of $m+n$. Let us take as an induction hypothesis that all the statements of the theorem are true when $m+n$ is decreased by 1 or more. The base of the induction is $m+n=2$. The only non-trivial case is $m=n=1$. We take $z^{(0,n)}=1$ and $z^{(m,n)}=0$ for $m>n$. Theorem 4.1 establishes the base case and also the first line of (d) for all n.

Induction step for (a), (b), (c), (d). Recall that (11) states

$$G_{m,n}(\mathbf{x}_m) = \theta\Big[g_{m,n-1}(x_m)F(x_m) + \int_{x_m}^{1} g_{m,n-1}(y)dF(y)\Big] + \theta\int_{x_m}^{1}\Big[g_{m-1,n-1}(x_m) - g_{m-1,n-1}(y)\Big]dF(y) - x_m + \Big\{V^{\mathrm{POST}}_{m-1,n}(\mathbf{x}_{m-1}) - V_{m-1,n}(\mathbf{x}_{m-1})\Big\}.$$

By the induction hypothesis for (f) we know that $z^{(m,n-1)} \leq z^{(m-1,n-1)}$. We now show that this inequality is strict and that $G_{m,n}(\mathbf{x}_{m-1}, x_m) = 0$ for a unique x_m, which lies in the interior of the interval $[z^{(m,n-1)}, z^{(m-1,n-1)}]$.

First, consider the case that x_m is above the upper endpoint of this interval, that is, suppose $x_m \geq z^{(m-1,n-1)}$. Let $y \geq x_m \geq z^{(m-1,n-1)}$. By the induction hypothesis for (f), we know that $y \geq z^{(m,n-1)}$ and this implies that $g_{m,n-1}(y) = g_{m-1,n-1}(y) = y$. The term in curly brackets in (11) is always non-positive, so

$$G_{m,n}(\mathbf{x}_m) \leq -(\theta-1)x_m < 0. \tag{12}$$

Now consider the case $x_m \leq z^{(m,n-1)}$. The term in curly brackets in (11) is 0, by the induction hypothesis for (b), and then by induction hypotheses for (f) and (d) we have

$$x_{m-1} \leq z^{(m,n-1)} \leq z^{(m-1,n-1)} < z^{(m-1,n)}.$$

Thus, $G_{m,n}(\mathbf{x}_m)$ depends only on x_m when $x_m < z^{(m,n-1)}$.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

Finally, consider $x = z^{(m,n-1)}$. By the induction hypothesis for (d) we know that $z^{(m,n-1)} > z^{(m,n-2)}$. So for all $y \geq z^{(m,n-1)}$ we have $g_{m,n-1}(y) = g_{m,n-2}(y) = y$. Using this fact, and the induction hypothesis for (g) we find:

$$\begin{aligned} G_{m,n}(\mathbf{x}_m) - G_{m,n-1}(\mathbf{x}_m) &= \theta \int_{x_m}^{1} \Big[g_{m-1,n-1}(x_m) - g_{m-1,n-1}(y) \Big] dF(y) \\ &\quad - \theta \int_{x_m}^{1} \Big[g_{m-1,n-2}(x_m) - g_{m-1,n-2}(y) \Big] dF(y) \\ &= \theta \int_{x_m}^{1} \int_{x_m}^{y} \Big[g'_{m-1,n-2}(\zeta) - g'_{m-1,n-1}(\zeta) \Big] d\zeta dF(y) \geq 0. \quad \textbf{(13)} \end{aligned}$$

From (12) and (13) we may deduce $z^{(m,n-1)} < z^{(m-1,n-1)}$. But knowing this, we see that the inequality in (13) is strict because $x_m = z^{(m,n-1)} < z^{(m-1,n-1)}$. Then by the induction hypothesis for (g) the integrand in (13) is strictly positive for values of y between x_m and $z^{(m-1,n-1)}$.

By Lemma 4.4, we know that $G_{m,n}(\mathbf{x}_{m-1}, x_m)$ is strictly decreasing and convex in x_m, and thus $G_{m,n}(\mathbf{x}_{m-1}, x_m) = 0$ has a unique root $x_m = z^{(m,n)}$ which does not depend on $\mathbf{x}_{m-1}$ and this root lies in the interval $\big(z^{(m,n-1)}, z^{(m-1,n-1)}\big)$. This establishes the induction step for (a), (b), and (d), and (c) follows immediately.

Induction step for (e). Consider a state $\mathbf{x}_m = (\mathbf{x}_{m-2}, x_{m-1}, x_m)$. We have from (c), as established immediately above, that

$$V_{m,n}(\mathbf{x}_m) = g_{m,n}(x_m) + g_{m-1,n}(x_{m-1}) + V_{m-2,n}(\mathbf{x}_{m-2}).$$

As explained in the proof of Lemma 4.3 (in Appendix C), we can interpret $g'_{m,n}(x_m)$ as the probability that job m is assigned to a resource under an optimal policy. Similarly, $g'_{m-1,n}(x_{m-1})$ is the probability that job $m-1$ is assigned to a resource under an optimal policy. But by Lemma 4.2, we know that an optimal policy will always assign job m before assigning job $m-1$, and so we may deduce that $g'_{m,n}(x_m) \geq g'_{m-1,n}(x_{m-1})$ for all $x_m \geq x_{m-1}$. (Note that these are right-hand derivatives and so our claim is correct because we are following a policy that assigns job m to a resource if $x_m = z^{(m,n)}$).

Induction step for (f). By using the fact just proved above for (e), and the fact that $g'_{m,n}(1) = g'_{m,n-1}(1) = 1$,

$$\int_{z^{(m-1,n)}}^{1} \Big[g'_{m,n}(x) - g'_{m-1,n}(x) \Big] dx = \big[1 - g_{m,n}(z^{(m-1,n)})\big] - \big[1 - z^{(m-1,n)}\big] \geq 0,$$

such that $g_{m,n}(z^{(m-1,n)}) \leq z^{(m-1,n)}$, and this implies $z^{(m,n)} \leq z^{(m-1,n)}$.

Induction step for (g). In the region $x_m < z^{(m,n-1)} < z^{(m,n)}$,

$$g_{m,n}(x_m) = V^{\mathrm{POST}}_{m,n}(\mathbf{x}_m) - V_{m-1,n}(\mathbf{x}_{m-1}).$$

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

Suppose $x_m = x$ and $x < z^{(m,n-1)} < z^{(m,n)}$, then

$$\begin{aligned} g'_{m,n}(x) &= \frac{d}{dx} V^{\text{POST}}_{m,n}(\mathbf{x}_m) \\ &= \frac{d}{dx}\left\{\theta\left[g_{m,n-1}(x)F(x) + \int_x^1 \Big[g_{m,n-1}(y) + g_{m-1,n-1}(x) - g_{m-1,n-1}(y)\Big]dF(y)\right]\right\} \\ &= \theta\Big[g'_{m,n-1}(x)F(x) + g'_{m-1,n-1}(x)\bar{F}(x)\Big] \qquad \textbf{(14)} \\ &< \theta\Big[g'_{m,n-2}(x)F(x) + g'_{m-1,n-2}(x)\bar{F}(x)\Big] \\ &= g'_{m,n-1}(x), \end{aligned}$$

where the inequality follows from the induction hypothesis for (g) and it is strict since $x < z^{(m,n-1)}$. If $z^{(m,n-1)} \le x < z^{(m,n)}$, then $g'_{m,n-1}(x) = 1$ and from (14) we see that $g'_{m,n}(x) < 1$ and so $g'_{m,n}(x) < g'_{m,n-1}(x)$. Finally, if $x \ge z^{(m,n)}$, we have $g'_{m,n}(x) = g'_{m,n-1}(x) = 1$. ■

COROLLARY 4.2.1: *As* $n \to \infty$, $z^{(m,n)} \to z^{(m)}$, *and* $g_{m,n}(x) \to g_m(x)$. *In particular,* $g_m(x) = z^{(m)}$ *for* $x < z^{(m)}$. *The expected profit is* $V^{\pi^*}_m(\boldsymbol{x}_m) = g_m(x_m) + V^{\pi^*}_{m-1}(\boldsymbol{x}_{m-1})$.

PROOF: First, from (d) of Theorem 4.2, we see that $z^{(2,n)} < z^{(1)}$, indicating that $z^{(2,n)}$ monotonically converges and $z^{(2)} \le z^{(1)}$. It follows by induction on m that $z^{(m)} \le z^{(m-1)}$. We next show that $g_m(x) = z^{(m)}$ for $x < z^{(m)}$. Note that the case of $m = 1$ is given in Lemma 4.1 and Theorem 4.1. Suppose that $g_{m-1}(x) = z^{(m-1)}$ for $x < z^{(m-1)}$. For $x < z^{(m)}$, we have

$$\begin{aligned} g_m(x) &= \theta\Big[g_m(x)F(x) + \int_x^1 g_m(y)dF(y)\Big] + \theta\int_x^1 \Big[g_{m-1}(x) - g_{m-1}(y)\Big]dF(y) \\ &= \theta\Big[g_m(x)F(x) + \int_x^1 g_m(y)dF(y)\Big] + \theta\int_{z^{(m-1)}}^1 \Big[z^{(m-1)} - y\Big]dF(y), \end{aligned}$$

where the second equality follows since $z^{(m)} \le z^{(m-1)}$ and $z^{(m-1)} = g_{m-1}(x) = g_{m-1}(y)$ for $x \le y \le z^{(m-1)}$. Therefore, $g'_m(x) = \theta g'_m(x)F(x)$ or $g'_m(x) = 0$. It is straightforward that $g_m(x) = z^{(m)}$ for $x < z^{(m)}$. ■

Thus, the threshold policy has been established for the SSAP with the postponement option and homogeneous resources. Intuitively, the benefit of the postponement option is to "improve" the queue $\mathbf{x}_m$ by updating it with the arrival of a new job. The number of resources to be allocated, m, reflects the effect of discounting. The decision-maker is more conservative in postponing when m increases. On the other hand, the number of jobs yet to arrive, n, measures the potential for improvement. The decision-maker is more selective when accepting a job as n increases. Moreover, the monotonicity of $g'_{m,n}(x)$ in m indicates that the improvement on x_m gives the largest profit increase.

4.2. The Benefits of Postponement

For the sake of comparison, we introduce the model for the SSAP without the postponement option, the study of which is abundant in the literature (e.g., Sections 2, 3 of Karlin [11], Derman, Lieberman, & Ross [6] and Sakaguchi [25]). The system state is denoted as $(x, \mathbf{p}_m, n)$, where x is the value of the job on hand. Decision epochs occur when jobs arrive. When a job valued at x arrives, if the decision-maker immediately accepts the job and

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

assigns it to a resource with quality value p_i, then a reward $p_i x$ is received. The state of the system then transitions to $\big(y, (p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_m), n-1\big)$, where y is the value of the next arriving job. If the decision-maker immediately rejects the job, the state transitions to $\big(y, \mathbf{p}_m, n-1\big)$. We define Π_{IM} as the set of policies that require an immediate decision (no postponement option) and ψ^* as the optimal policy in Π_{IM}. Additionally, we define $D^{\psi^*}_{m,n}(x, \mathbf{p}_m)$ as the decision rule and $V^{\psi^*}_{m,n}(x, \mathbf{p}_m)$ as the expected profit of state $(x, \mathbf{p}_m, n)$. Denote the threshold value used in ψ^* as $w^{(m,n)}$. Unlike the situation with the postponement option, with policy ψ^*, only one decision occurs at a time, since no queue of jobs is maintained. Indeed, it can be shown that as n goes to infinity, $w^{(m,n)}$ and $V^{\psi^*}_{m,n}(x)$ converge. We denote the limits as $w^{(m)}$ and $V^{\psi^*}_{m}(x)$, respectively.

Below, we summarize the results for the SSAP without the postponement option (mainly attributable to [2,6,25]). As we assume homogeneous resources in this section, we remove $\mathbf{p}_m$ from the state definition.

THEOREM 4.3: *Given an arriving job valued at x, m remaining homogeneous resources, n jobs yet to arrive, and $m \le n$, the optimal decision, $D^{\psi^*}_{m,n}(x)$, is to immediately accept the job and assign a resource to it if $x \ge w^{(m,n)}$ or to immediately reject the job if $x < w^{(m,n)}$. The threshold value $w^{(m,n)}$ is*

$$w^{(m,n)} = g_{1,1}(w^{(m,n-1)}) + \theta \sum_{k=1}^{m-1} w^{(k,n-1)} - \sum_{k=1}^{m-1} w^{(k,n)}, \tag{15}$$

where $w^{(m,k)} = 0$ for $k < m$ and $w^{(0,n)} = 0$. The corresponding expected profit is

$$V^{\psi^*}_{m,n}(x) = \begin{cases} w^{(m,n)} + \sum_{k=1}^{m-1} w^{(k,n)}, & x \in [0, w^{(m,n)}) \\ x + \sum_{k=1}^{m-1} w^{(k,n)}, & x \in [w^{(m,n)}, 1] \end{cases}. \tag{16}$$

Moreover, $w^{(m+1,n)} \le w^{(m,n)} \le w^{(m,n+1)}, \forall m \ge 1, n \ge 1$.

Compared with policy ψ^*, the advantage of allowing postponement with policy π^* is obvious for finite n. Intuitively, given an arriving job valued at x, m, and n, the job should be rejected if $x \in (w^{(m,n-1)}, w^{(m,n)})$, according to ψ^*. However, it is likely that the next arriving job is valued less than $w^{(m,n-1)}$ and in this case the decison-maker may regret rejecting the job valued at x. The postponement option hedges against this risk. Also, since $\Pi_{\mathrm{IM}} \subseteq \Pi_{\mathrm{P}}$, $V_{m,n}(0, \ldots, 0, x) \ge V^{\psi^*}_{m,n}(x)$. For infinite n, even with the postponement option, the decision-maker will never accept jobs retained in the queue, since their values are less than $z^{(m)}$ (see Corollary 4.2.1). Thus, there is no benefit to keeping jobs in the queue and the postponement option has no benefit. We summarize the results in the following theorem.

THEOREM 4.4: *With finite m and n, $V_{m,n}(0, \ldots, 0, x) \ge V^{\psi^*}_{m,n}(x)$. Specifically, policy π^* strictly outperforms policy ψ^* under the following conditions: (1) $m = n$ and $x \in (0, z^{(m,m)})$ (with $z^{(1,1)} = z^{(1)}$), (2) $n > m = 1$ and $x \in [0, z^{(1)})$, or (3) $n > m \ge 2$. Finally, in the case of infinite n, $z^{(m)} = w^{(m)}$ and $V_m(0, \ldots, 0, x) = V^{\psi^*}_{m}(x)$.*

PROOF: See Appendix B. ■

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

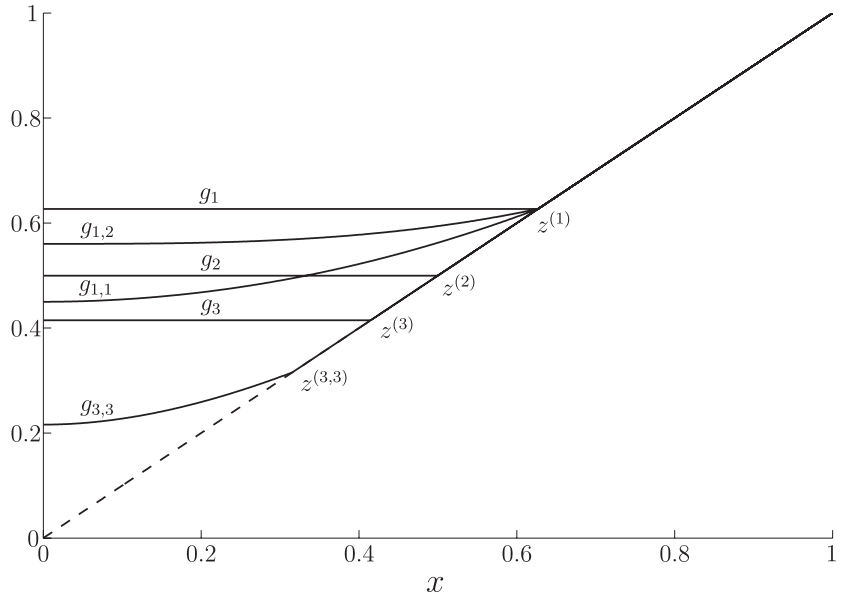


FIGURE 1. Properties in Theorem 4.2, illustrated when $X \sim \text{Beta}(1,1)$.

TABLE 1. Threshold Value Differences: $z^{(m,n)} - w^{(m,n)}$, where $X \sim 10\text{Beta}(2,5)$ (E[X] = 20/7)

m,n	1	2	3	4	5	6	7	8	9	10
1	4.1	3.4	3.0	2.7	2.5	2.3	2.2	2.0	1.9	1.8
2	0.0	2.9	2.6	2.4	2.2	2.0	1.9	1.8	1.7	1.6
3	0.0	0.0	2.2	2.3	2.1	1.9	1.8	1.7	1.6	1.5
4	0.0	0.0	0.0	1.8	1.7	1.9	1.7	1.6	1.5	1.4
5	0.0	0.0	0.0	0.0	1.5	1.5	1.5	1.5	1.5	1.4

4.3. Numerical Example

To illustrate the benefits of postponement, experiments are conducted for rewards $X \sim a\xi$, where ξ is a beta-distributed random variable with a p.d.f. proportional to $\xi^{\alpha-1}(1-\xi)^{\beta-1}$, where α and β are parameters. We numerically compute functions $g_{m,n}(x)$ and $V^{\psi^*}_{m,n}(x)$ and threshold values $z^{(m,n)}$ and $w^{(m,n)}$. Moreover, to study the variability of the profits, we conduct simulation experiments based on the computed threshold values; specifically, we generate job arrivals (by generating the beta-distributed values and Poisson arrivals through Excel) and apply policies π^* and ψ^*.

Assume that $\lambda = 1$, $\gamma = 1/9$, and that $X \sim \text{Beta}(1,1)$. Figure 1 illustrates the properties of $g_{m,n}(x)$ and $z^{(m,n)}$. It shows that as n grows, $g_{m,n}(x)$ become "flatter" for small x values. Also, the benefit of an increase in n drops with n. Moreover, when $m = n$, $g_{m,n}$ decreases with m.

Now assume the same λ and γ and $X \sim 10\text{Beta}(2,5)$. As shown in Table 1, the numerical results indicate that $z^{(m,n)} \geq w^{(m,n)}$. Also, as m grows, the decision-maker is more conservative in postponement and the differences between the threshold values of the two policies narrow, defining a decrease in the benefit of postponement. Figure 2 shows the percentage of improvement in the expected profit of the postponement option given the above parameters for various m and n values.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

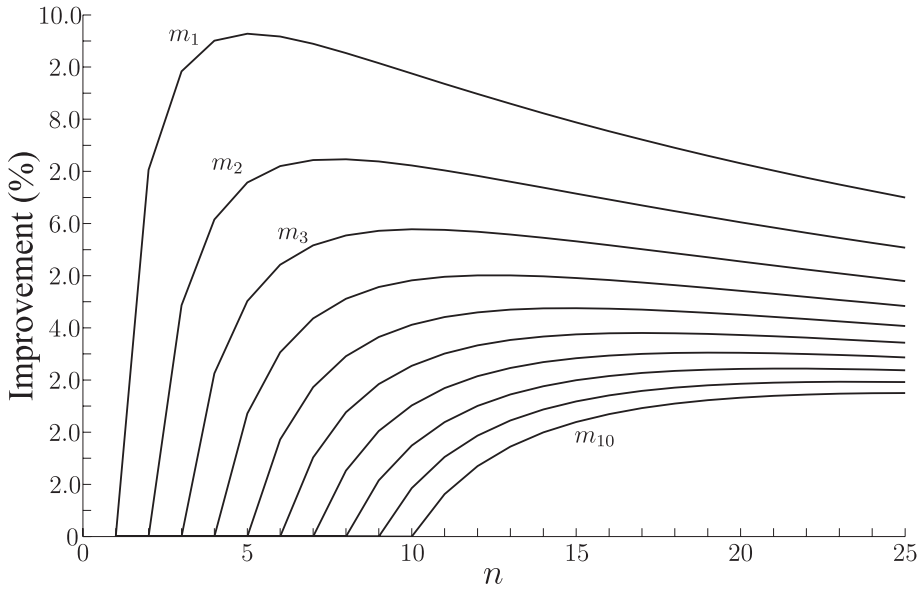


FIGURE 2. Discounted profit improvement with $X \sim 10\text{Beta}(2, 5)$ (left skewed F, homogeneous resources).

Another interesting observation is that the benefit of postponement is not monotone in n. Intuitively, when the number of jobs that can be queued is small, so is the chance of postponement. As n grows, the decision-maker can retain more jobs in the queue and the benefit of postponement increases. However, as n grows further, the benefit of postponement decreases with $n \to \infty$, as discussed in Section 4.2.

Figure 3 illustrates the percentage of improvement in the expected profit with the same parameters, except that $X \sim 10\text{Beta}(5, 2)$. Note that the postponement option is more valuable when the distribution of X is skewed to the left (when $E[X]$ is smaller). Intuitively, this is because in the latter case most jobs have values close to the maximum possible value of 10. So there is a little scope for gaining any improvement by postponing jobs. Whereas, in the former case, there will be occasional jobs that are substantially more valuable than the average job and so it is worth making postponement to wait for these.

Finally, the potential of the postponement option in hedging against undesirable arriving jobs may lead to a lower variability of the profit. To illustrate, consider the first parameter settings introduced above and let $m = 9$ and $n = 20$. Based on 10,000 runs of policy π^*, the average profit is 37.72, with a sample standard deviation of 4.39, giving a 95% confidence interval for the average profit of $37.72 \pm 1.96 \times 4.39/100$, or $[37.63, 37.81]$. For policy ψ^*, the average profit is 36.65, with a sample standard deviation of 4.51, giving a 95% confidence interval for the average profit of $[36.56, 36.74]$.

5. HETEROGENOUS RESOURCES

With homogeneous resources, the postponement option reduces the regret of rejecting a "valuable" job or accepting a "less valuable" job, compared with jobs arriving later. With heterogeneous resources, this benefit may be amplified by the difference in resource values. However, the SSAP with heterogeneous resources is much harder to analyze. In this section, we first analyze the optimal policies with and without postponement, respectively denoted

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

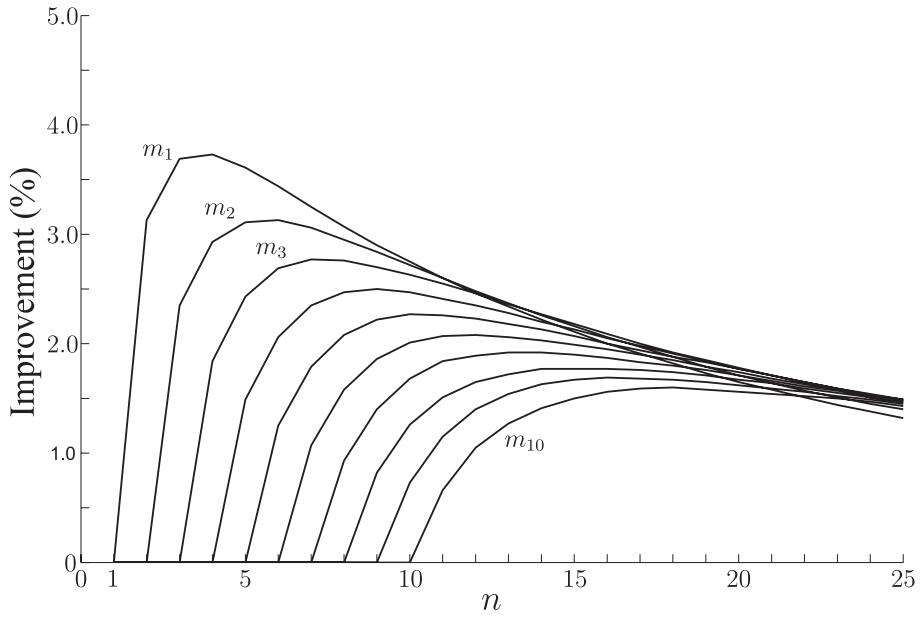


FIGURE 3. Discounted profit improvement with $X \sim 10\text{Beta}(5, 2)$ (right skewed F, homogeneous resources).

by π^* and ψ^*, and identify the difference between them. Then, we propose two heuristics for the SSAP with heterogeneous resources, based on the insights obtained from the analysis on the SSAP with homogeneous resources. Additionally, we use numerical experiments to illustrate the performance of the heuristics. It should be noted that the case of $m = 1$ is trivial. In this section, we only focus on the case of $m \geq 2$.

5.1. Properties

Consider the optimal policy $\psi^* \in \Pi_{\text{IM}}$ with heterogeneous resources. The literature on SSAP (without the postponement option) shows that policy ψ^* follows a simple threshold structure, as described at the beginning of Section 2. The existence of the threshold structure for policy ψ^* can be interpreted as follows. Given a job with value x, m resources and n jobs yet to arrive, if the decision-maker assigns the job to a resource, then there are $m - 1$ jobs to be assigned in the future and the decision-maker should compare x with the expected values (after discounting) of these $m - 1$ jobs. Define these values as $\mathbb{E}(X_{m-1}^{(i,n)})$, $i = 1, 2, \ldots, m-1$, with $\mathbb{E}(X_{m-1}^{(i,n)})$ as the ith largest one. With policy ψ^*, it turns out that $\mathbb{E}(X_{m-1}^{(i,n)}) = w^{(i,n)}$. Thus, the optimal decision follows the rearrangement inequality. For example, if $x > w^{(1,n)}$, it is optimal to assign this job to resource m. Below, we rephrase this result in Theorem 5.1.

THEOREM 5.1: *Given $\boldsymbol{p}_m = (p_1, \ldots, p_m)$, n jobs yet to arrive and a job valued at x, the optimal decision is*

$$D_{m,n}^{\psi^*}(x) = \begin{cases} \text{reject}, & x \leq w^{(m,n)} \\ \text{assigntoresource}i, & w^{(m-i+1,n)} < x \leq w^{(m-i,n)}, i = 1, 2, \ldots, m-1 \\ \text{assigntoresource}m, & x > w^{(1,n)} \end{cases}, \quad \textbf{(17)}$$

and $\mathbb{E}(X_m^{(i,n)}) = w^{(i,n)}$, for $i = 1, 2, \ldots, m$.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

Unfortunately, policy $\pi^* \in \Pi_P$ does not possess a simple threshold structure when resources are heterogeneous. The updating rule U remains the same, while the acceptance and assignment rule A^* is complicated in this case. First, consider the optimal decisions under extreme cases. Given a state $(\mathbf{x}_m, \mathbf{p}_m)$ and n jobs yet to arrive, if $\mathbf{x}_m = (0, \ldots, 0)$, then to immediately accept a job and assign it to a resource is equivalent to abandoning this resource, and thus is not optimal. If $\mathbf{x}_m = (0, \ldots, 0, 1)$, there is no benefit to postpone due to discounting. Thus, it is optimal to immediately accept a job with a high value and assign it to a resource, as in the following theorem.

THEOREM 5.2: *Given $\boldsymbol{x}_m, \boldsymbol{p}_m$, and n, and $x_{i-1} < z^{(1)} \leq x_i$, the optimal decision accepts job i, $i+1$, …, m and assigns job i to resource i.*

PROOF: See Appendix C. ■

Despite these cases, it is difficult to find an optimal policy with a simple threshold structure as in ψ^*. The decision of whether to postpone indeed depends on both $\mathbf{x}_m$ and $\mathbf{p}_m$. Consider the situation where $\mathbf{x}_2 = (x_1, x_2)$, $\mathbf{p}_2 = (p_1, p_2)$, and $n = 1$, where $x_1 = 0$ and $x_2 > 0$. With policy π^*, the expected profit is:

$$V_{2,1}(\mathbf{x}_2, \mathbf{p}_2) = \max\Bigg\{p_1 x_2 + p_2 \theta\mu, \quad p_1\theta\mu + p_2 x_2, \\ \theta p_1\Big[\int_{y \leq x_2} y dF(y) + x_2 \bar{F}(x_2)\Big] + \theta p_2\Big[x_2 F(x_2) + \int_{y > x_2} y dF(y)\Big]\Bigg\}.$$

Besides the first two expressions in the curly brackets, which can also be achieved by policy ψ^*, the third expression is the expected profit of postponement. It can be readily verified that whether to postpone depends on the values of p_1 and p_2. Next, consider the same situation except that $x_1 > 0$ and $x_2 < z^{(1)}$. There is no guarantee that job 2 should be assigned first (unlike the case with homogeneous resources). For example, if p_1 is nearly zero and p_2 is very large, it is possible that to immediately assign job 1 to resource 1 yields a higher profit than to immediately assign job 2 to resource 1 or resource 2, if x_1 is very small. Moreover, it can be seen from these two examples that the domain of the expected profit function is divided into several regions, each corresponding to an optimal decision. It is difficult to compute the optimal policy.

5.2. Heuristics for Heterogeneous Resources

In the literature, there are generally two distinctive methodologies to tackle a stochastic problem like the SSAP when a threshold policy does not exist: algorithms associated with MDPs and online algorithms. The algorithms associated with MDPs generally define states and use an iterative approach (e.g., Puterman [18] and Powell [19]) to solve the dynamic programming equation. Online algorithms generally use heuristics to estimate the performance of each action at a decision epoch (e.g., Hentenryck and Bent [9]), without defining states and solving the dynamic programming equation. There are also algorithms combining these two approaches (e.g., Ilhan, Iravani, & Daskin [10]). In this section, we use the second approach to build a benchmark policy illustrated below.

Define the benchmark policy, $\varphi \in \Pi_P$, which considers two actions at each decision stage: (1) assign resource i to job j, $1 \leq i, j \leq m$; (2) wait for another job, respectively. The expected profit of following each action is estimated (approximated) as the summation of

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

the reward yielded by this action and the expected profit of following ψ^* thereafter. The decision is to take the action generating the highest estimated profit.

We design a simulation experiment with 10,000 sample runs. Parameter values are set as: $\lambda = 1, \gamma = 0.001, X \sim 10\text{Beta}(2,5), m = 9$, and $n = 20$, and the values of $p_i = 0.2 + 0.2i, i = 1, 2, \ldots, 9$, or

$$(p_1, p_2, \ldots, p_9) = 0.2 \times (1, 2, \ldots, 9).$$

For policy ψ^*, the sample mean and the sample standard deviation of the average profit are 47.55 and 6.31, respectively. For policy φ, the corresponding values are 48.91 and 6.05, respectively. The profit improvement is 2.86%

Another way of measuring the performance of the heuristic policies is to study the upper bound of $V_{m,n}$. Define Y_j as the discounted value of job j that arrives at time T_j with value X_j (i.e., $Y_j = e^{-\gamma T_j} X_j$). Moreover, define $Y^{(i,n)}$ as the ith largest value in $\{Y_j\}_{j=1}^n$. Then, the ideal assignment (assuming $m \le n$) yields $\sum_{i=1}^m p_i Y^{(m-i+1,n)}$. Obviously, this profit is not achievable in the context of the SSAP, for it is not possible to decide $Y^{(i,n)}$, before all jobs arrive. Considering Theorem 5.1, we have

$$\sum_{i=1}^{m} p_i w^{(m-i+1,n)} \le V_{m,n}(\mathbf{0}, \mathbf{p}_m) \le \sum_{i=1}^{m} p_i \mathbb{E}[Y^{(m-i+1,n)}]. \tag{18}$$

The value of $\mathbb{E}[Y^{(m-i+1,n)}]$ can be computed through simulation and sampling. In the above parameter settings, the upper bound is $\sum_{i=1}^m p_i \mathbb{E}[Y^{(m-i+1,n)}] = 49.56$, defining a maximum profit improvement of 4.23%.

5.2.1. Heuristic 1: raising thresholds. In this section, we design a policy $\tilde{\pi} \in \Pi_P$, with a similar structure to policy ψ^* as described in Theorem 5.1 but with higher thresholds. We show that policy $\tilde{\pi}$ yields a higher expected profit than policy ψ^*. In this section, we only focus on the case of $2 \le m \le n$.

Specifically, with policy $\tilde{\pi}$, the decision rule for $m \le n$ is defined as:

$$D_{m,n}^{\tilde{\pi}}(\mathbf{x}_m) = \begin{cases} \text{postpone}, & x_m \le v_{m-1}^{(m,n)} \\ \text{acceptandassignjobmtoresourcei}, & v_{m-1}^{(m-i+1,n)} < x_m \le v_{m-1}^{(m-i,n)}, \\ & i = 1, 2, \ldots, m-1 \\ \text{acceptandassignjobmtoresourcem}, & x_m > v_{m-1}^{(1,n)} \end{cases} \tag{19}$$

where $v_{m-1}^{(m,n)} = w^{(m,n)}$ and $v_{m-1}^{(i,n)}$ with $1 \le i \le m-1$ bears the same meaning of $\mathbb{E}[X_{m-1}^{(i,n)}]$ except that policy $\tilde{\pi}$ is followed and the initial state is $\big((0, \ldots, 0), \mathbf{p}_{m-1}\big)$. In particular, $v_1^{(1,n)} = g_{1,n}(0)$.

THEOREM 5.3: *For* $2 \le m \le n$, $V_{m,n}^{\tilde{\pi}}\big((0, \ldots, 0, x_m), \boldsymbol{p}_m\big) \ge V_{m,n}^{\psi^*}(x_m, \boldsymbol{p}_m)$ *and* $v_m^{(i,n)} \ge w^{(i,n)}$ *for* $i = 1, 2, \ldots, m$, *where* n *is finite and the inequality is strict for* $m < n$. *Also,* $v_m^{(m,n)}$ *is independent of* $\boldsymbol{p}_m$.

PROOF: See Appendix D. ■

Recall that for the homogeneous resources situation, policy π^* and policy ψ^* are equivalent as $n \to \infty$. Based on this result and the decomposition approach introduced in Theorems 4 and 5 of Albright [2], it follows that the equivalence between policy π^* and

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

TABLE 2. Average Values of Jobs Assigned to Resources Without and With the Postponement Option

Resources	ψ^*	$\tilde{\pi}$	$\pi^\dagger$	ψ^* vs $\tilde{\pi}$(%)	ψ^* vs $\pi^\dagger$(%)
p_9	5.64	5.66	6.06	0.36	7.45
p_8	5.00	5.04	5.30	0.80	6.00
p_7	4.55	4.61	4.79	1.17	5.27
p_6	4.22	4.23	4.37	0.39	3.55
p_5	3.91	3.96	4.01	1.12	2.56
p_4	3.64	3.69	3.72	1.29	2.20
p_3	3.39	3.48	3.46	2.72	2.06
p_2	3.18	3.38	3.21	6.30	0.94
p_1	2.97	3.36	2.98	12.93	0.34

policy ψ^* also holds for the heterogeneous resources situation as $n \to \infty$. Thus, policy $\tilde{\pi}$ converges to policy π^*, as stated in the following theorem:

THEOREM 5.4: *For $m \geq 2$ and heterogeneous resources,*

$$\lim_{n\to\infty} V_{m,n}(\mathbf{0}, \boldsymbol{p}_m) = \lim_{n\to\infty} V^{\tilde{\pi}}_{m,n}(\mathbf{0}, \boldsymbol{p}_m) = \lim_{n\to\infty} V^{\psi^*}_{m,n}(0, \boldsymbol{p}_m).$$

Note that, it is not easy to compute $v_m^{(i,n)}$. A reasonable approximation is a lower bound:

$$\Big\{ v_m^{(i,n-1)} F(w^{(m,n)}) + v_{m-1}^{(i,n-1)} \Big[F(v_{m-1}^{(i,n-1)}) - F(w^{(m,n)}) \Big] + \int_{v_{m-1}^{(i,n-1)}}^{v_{m-1}^{(i-1,n-1)}} y dF(y) + v_{m-1}^{(i-1,n-1)} \bar{F}(v_{m-1}^{(i-1,n-1)}) \Big\},$$

which is the right hand side of (20) (see Appendix D for a detailed explanation). Numerically, higher profits can be obtained by applying even higher thresholds. As can be expected, the profit follows a parabolic path with the increasing thresholds; raising the thresholds too much usually hurts the profit due to the discount factor. Through trial-and-error, we found that the potential for increasing $v_m^{(i,n-1)}$ grows significantly with i and moderately with n/m. Through experimentation, we added the term $0.02\sqrt{(n/m)}3^{0.15i}$ to the right-hand side of (20). Following this policy, the sample mean and the sample standard deviation of the average profit are 48.53 and 6.20, respectively. Policy $\tilde{\pi}$ leads to a profit improvement 2.06% compared with policy ψ^*, but is inferior to the benchmark policy. By checking the average discounted values of jobs assigned to each resource, it can be seen that policy $\tilde{\pi}$ mainly improves the discounted values of the jobs assigned to the resources with low quality values (Table 3).

5.2.2. Heuristic 2: sampling on future arrivals. As illustrated, a policy similar to ψ^* with increased threshold values is not quite satisfactory. In this section, we define a policy $\pi^\dagger \in \Pi_P$ that utilizes $z^{(m,n)}$ and samples future arrivals. Also, we mainly focus on the case of $2 \leq m \leq n$.

Following decision rule $D^{\pi^\dagger}_{m,n}$, the first step in a decision is to see whether to postpone, depending on whether x_m is larger than $z^{(m,n)}$. If this is not the case, the decision-maker waits for the next arrival. If this is the case, the decision-maker accepts the queued job with the smallest value. The second step is to determine which resource to assign the accepted

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

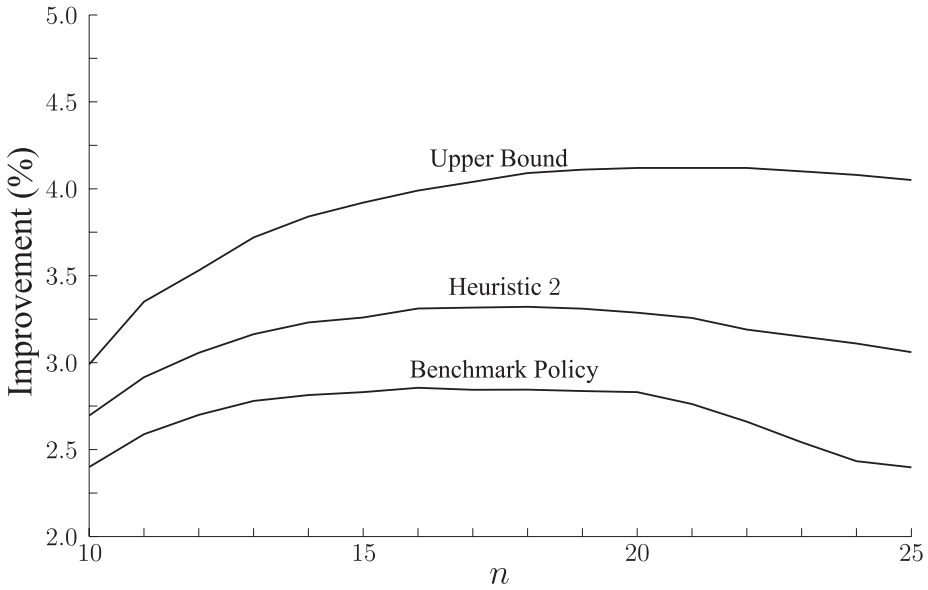


FIGURE 4. Discounted profit improvement with $X \sim 10\text{Beta}(2,5)$ (left skewed F, heterogeneous resources, and Heuristic 2).

job. The assignment rule follows a similar structure to D^{ψ^*} (see Theorem 5.1), except that $w^{(i,n)}$ are replaced by $\eta w^{(i,n)} + (1-\eta)\mathbb{E}[Y^{(i,n)}]$ with $0 \le \eta \le 1$.

Intuitively, $\eta w^{(i,n)} + (1-\eta)\mathbb{E}[Y^{(i,n)}]$ is an approximation of the expected discounted job value that is assigned to the resource with the ith largest quality value by following policy $\pi^\dagger$. For $\eta = 1$, $\eta w^{(i,n)} + (1-\eta)\mathbb{E}[Y^{(i,n)}] = w^{(i,n)}$ and policy $\pi^\dagger$ is very similar to policy ψ^*. For $\eta = 0$, $\eta w^{(i,n)} + (1-\eta)\mathbb{E}[Y^{(i,n)}] = \mathbb{E}[Y^{(i,n)}]$ and policy $\pi^\dagger$ performs poorly, since $\mathbb{E}[Y^{(i,n)}]$ is the ideal expected discounted job value assigned to the job with the ith largest quality value, which is not achievable. For some $\eta \in (0,1)$, policy $\pi^\dagger$ can achieve a higher profit. It should be clear that by simply applying this assignment rule alone, without the postponement option, policy $\pi^\dagger$ is inferior to policy ψ^*.

It can be seen that policy $\pi^\dagger$ utilizes the insights from policy π^* in the situation of homogeneous resources. First, the condition of whether x_m is smaller than $z^{(m,n)}$ serves as a threshold of postponement. Second, when accepting jobs, policy $\pi^\dagger$ chooses the queued job with the smallest value. The intuition behind is that to retain job m in the queue may yield a larger profit, as indicated by the monotonicity of $g'_{m,n}(x^+)$ in m (see Theorem 4.2).

Under the parameter settings from Section 5.2, this policy significantly outperforms the first heuristic policy. With $\eta = 0.55$, the sample mean and the sample standard deviation (over 10,000 runs) of the expected profit are 49.13 and 6.00, respectively. Policy $\pi^\dagger$ yields a profit improvement of 3.32%. The 95% confidence intervals of policy ψ^* and $\pi^\dagger$ are $[47.42, 47.67]$ and $[49.01, 49.25]$, respectively. And it can be verified through a F-test that policy $\pi^\dagger$ reduces the variability of the profit. Indeed, policy $\pi^\dagger$ dominates the benchmark policy for a variety of n values, as illustrated in Figure 4.

Though it is hard to evaluate the performance of policy $\pi^\dagger$ in terms of the closeness to optimality, we can better understand its performance through the following example. Consider the case of $\mathbf{x}_2 = (0, x_2), \mathbf{p}_2 = (p_1, p_2)$ and $n = 1$, in which we can find the optimal policy, π^*. The profit improvement increases as $p_2 - p_1$ grows. Intuitively, the expected

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

discounted value of the job assigned to resource 2 under policy π^* is larger than its counterpart under policy ψ^*, while the expected discounted value of the job assigned to resource 1 is smaller than its counterpart under policy ψ^*. In other words, policy π^* achieves a higher profit since it utilizes additional information such that resources with higher values are assigned to better jobs. Indeed, Table 4 illustrates that policy $\pi^\dagger$ achieves the same objective, though not to the optimal extent.

6. CONCLUSIONS

In this paper, the SSAP has been extended to include the postponement option. Specifically, this case allows for the analysis of problems where the decision-maker can delay the decision to accept or reject a job upon arrival for some period of time. The tradeoff to be considered is the value of delaying the decision, allowing more information to be gathered, against the decline in value of accepting a job late, due to discounting. There are a variety of applications where an assignment can be delayed in order to gather more information. For example, most interviewing processes allow for a delay before offers are made and/or accepted. Similarly, multiple bids for the sale of an item or property may be received before acceptance.

It was shown that the optimal policy of the SSAP with homogeneous resources has a threshold structure. Despite defining the state space according to a vector, the analysis merely requires consideration of the maximum valued job on hand. The situation with heterogeneous resources was also analyzed, which is significantly more difficult. As the analysis must consider both the values of the jobs and resources simultaneously, it is difficult to find an optimal policy possessing a simple threshold structure. Two heuristic policies are proposed, based on insights derived from the case with homogeneous resources. Numerically, policy $\tilde{\pi}$, which adapts the threshold policy found in the SSAP literature (i.e., policy ψ^*) by raising threshold values does not work well. Policy $\pi^\dagger$, which samples future job arrivals and employs the threshold $z^{(m,n)}$ derived for the homogeneous resources case achieves the highest profit.

We believe there are significant extensions of this work that are worthy of consideration. Note that nothing about the proofs in Section 4 would differ in any essential way if we were to add to our model a clock that inserts extra decision times as a Poisson process of rate μ. These are purely fictional times, at which nothing takes place. The only difference is that with probability $\mu/(\lambda+\mu+\gamma)$ no job may arrive between one decision time and the next. However, this is a trivial change to incorporate into the analysis. Now let the rate λ, of the Poisson stream of arriving jobs, depend on the number of decision times introduced by the clock to date, say k (i.e., when the clock reads k). The effect is to create a different value of θ, say $\theta_k = (\lambda_k+\mu)/(\lambda_k+\mu+\gamma)$, as the process evolves forward in time from a point at which the clocks reads k. Proofs obviously go through just as before, because we never use the fact that θ is the same at all decision times. By letting $\mu \to \infty$, we see that the type of optimal policy that is described in Theorem 4.2 is also valid if the job arrival stream is a non-homogeneous Poisson process, except that the threshold $z^{(m,n)}$ is now time-dependent, say $z^{(m,n)}(t)$.

Similarly, there is no difficulty in letting the discount rate be (clock) time-dependent, since again all this does is to make θ (clock) time-dependent. As a special case, let the discount rate be constant in $[0,T)$ and then infinite at T. We can deduce that in a problem, with discounting, in which the aim is to maximize the expected profit by T, the optimal decision rule is also one in which it is optimal to accept the job with greatest value if and only if this value exceeds some threshold $z^{(m,n)}(t)$, which depends on t, but, again surprisingly, not on any lesser values of jobs that are held in the postponement queue.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

Clearly, the case with heterogeneous assets can be explored further, as would be the case when resources become available over time (according to another stochastic process). Additionally, while the delay option may be feasible in a number of settings, there may be a limit as to how long a delay can occur. Thus, it would be interesting to study this problem in light of a deadline. In fact, it has been found through numerical experiments that $g_{k,m}(x)$ may not be significantly higher than $w^{(k,m)}$ if x is small, indicating that some jobs with low values in the queue do not contribute much and thus can be removed from the queue. In other words, it is likely to achieve a higher profit with moderate level of delay.

Acknowledgements

The authors gratefully acknowledge the support from NSF Grant CMMI-0813671 and CMMI-1100765 and the help from a dedicated reviewer.

References

1. Albright, C. & Derman, C. (1972). Asymptotic optimal policies for the stochastic sequential assignment problem. *Management Science* 19: 46–51.
2. Albright, S.C. (1974). Optimal sequential assignments with random arrival times. *Management Science* 21(1): 60–67.
3. Albright, S.C. (1976). A Markov chain version of the secretary problem. *Naval Research Logistics Quarterly* 1: 151–159.
4. Albright, S.C. (1977). A Bayesian approach to a generalized house selling problem. *Management Science* 24(4): 432–440.
5. Amram, M. & Kulatilaka, N. (1999). Real options: managing strategic investment in an uncertain world. Boston: Harvard Business School Press.
6. Derman, C., Lieberman, G.J., & Ross, S.M. (1972). A sequential stochastic assignment problem. *Management Science* 7: 349–355.
7. Elfving, G. (1967). A persistency problem connected with a point process. *Journal of Applied Probability* 4: 77–89.
8. Hardy, G.H., Littlewood, J.E., & Pólya, G. (1934). Inequalities. Cambridge: The University Press.
9. Hentenryck, P. & Bent, R. (2006). Online stochastic and combinatorial optimization. Cambridge, MA: The MIT Press.
10. Ilhan, T., Iravani, S.M.R., & Daskin, M.S. (2011). Technical note—the adaptive knapsack problem with stochastic rewards. *Operations Research* 1: 242–248.
11. Karlin, S. (1962). Stochastic models and optimal policy for selling an asset. Studies in Applied Probability and Management Science, K. J. Arrow, S. Karlin, H. Scarf, (eds.), Stanford, CA: Stanford University Press.
12. Kennedy, D.P. (1986). Optimal sequential assignment. *Mathematics of Operations Research* 4: 619–626.
13. Kleywegt, A.J. & Papastavrou, J.D. (1998). The dynamic and stochastic knapsack problem. *Operations Research* 46(1): 17–35.
14. McLay, L.A., Jacobson, S.H., Nikolaev, A.G. (2009). A sequential stochastic passenger screening problem for aviation security. *IIE Transactions* 41(6): 575–591.
15. Nakai, T. (1986a). A sequential stochastic assignment problem in a stationary Markov chain. *Mathematica Japonica* 31: 741–757.
16. Nakai, T. (1986b). A sequential stochastic assignment problem in a partially observable Markov chain. *Mathematics of Operations Research* 11: 230–240.
17. Nikolaev, A.G. & Jacobson, S.H. (2010). Technical note — stochastic sequential decision-making with a random number of jobs. *Operations Research* 58(4): 1023–1027.
18. Puterman, M.L. (1994). Markov decision processes: discrete stochastic dynamic programming. New Jersey: John Wiley & Sons.
19. Powell, W. (2007). Approximate dynamic programming: solving the curses of dimensionality. New Jersey: John Wiley & Sons.
20. Righter, R.L. (1987). The stochastic sequential assignment problem with random deadlines. *Probability in the Engineering and Informational Sciences* 1: 189–202.
21. Righter, R.L. (1989). A resource allocation problem in a random environment. *Operations Research* 37(2): 329–337.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

22. Ross, S.M. (1983). Introduction to stochastic dynamic programming. New York: Academic Press.
23. Saario, V. (1985). Limiting properties of the discounted house-selling problem. *European Journal of Operational Research* 20(2): 206–210.
24. Sakaguchi, M. (1984a). A sequential stochastic assignment problem associated with a non-homogeneous Markov process. *Mathematica Japonica* 29(1): 13–22.
25. Sakaguchi, M. (1984b). A sequential stochastic assignment problem associated with unknown number of jobs. *Mathematica Japonica* 29(2): 141–152.
26. Su, X. & Zenios, S. (2005). Patient choice in kidney allocation: a sequential stochastic assignment model. *Operations Research* 53(3): 443–455.
27. Triantis, A.J. (2000). Real options and corporate risk management. *Journal of Applied Corporate Finance* 13(2): 64–73.
28. Trigeorgis, L. (1996). Real options: managerial flexibility and strategy in resource allocation. Cambridge, MA: MIT Press.
29. Van Hoek, R.I. (2001). The rediscovery of postponement a literature review and directions for research. *Journal of Operations Management* 19(2): 161–184.
30. Zenios, S.A., Chertow, G.M., & Wein, L.M. (2000). Dynamic allocation of kidneys to candidates on the transplant waiting list. *Operations Research* 48(4): 549–569.

APPENDIX A: PROOF OF LEMMA 4.1

LEMMA 4.1: *Let $z^{(1)}$ be the value of x such that $x = \theta\mathbb{E}\big[\max\{x, X\}\big]$. The function sequence defined in (5) possesses the following properties:*

1. *$g_{1,n}(x)$ is strictly increasing in x for $x \in [0, z^{(1)})$ and $n = 1, 2, \ldots$;*
2. *$g_{1,n}(x)$ is strictly increasing in n for $x \in [0, z^{(1)})$;*
3. *$g_{1,n}(x) > x$ for $x \in [0, z^{(1)})$ and $g_{1,n}(z^{(1)}) = z^{(1)}$ for $n = 1, 2, \ldots$;*
4. *$1 > g'_{1,n}(x) > g'_{1,n+1}(x)$ for $x \in (0, z^{(1)})$ and $n = 1, 2, \ldots$;*
5. *$g_{1,n}(x) \to z^{(1)}$ as $n \to \infty$ for $x \in [0, z^{(1)})$;*
6. *$g'_{1,n}(x)$ is non-decreasing in x for $x \in (0, 1)$ and $n = 1, 2, \ldots$.*

PROOF: It is easy to see by induction that $g_{1,n}(z^{(1)}) = z^{(1)}$ for all n, and also that $g_{1,n}(x) = x$ for $x \geq z^{(1)}$. Also, $\theta\mathbb{E}\big[g_{1,n-1}\big(\max(x, X)\big)\big]$ is strictly convex in x, and so $g_{1,n}(x) > x$ for $x < z^{(1)}$. Property 3 is established.

From this, it follows that for $x < z^{(1)}$, $g_{1,n}(x) = \theta\mathbb{E}\big[g_{1,n-1}\big(\max(x, X)\big)\big]$, and we can conclude that:

$$g_{1,n}(x) = \begin{cases} \theta\Big(g_{1,n-1}(x)F(x) + \int_x^1 g_{1,n-1}(y)dF(y)\Big), & x < z^{(1)} \\ x, & x \geq z^{(1)} \end{cases}.$$

From the above expression, Property 3 follows. Moreover, for $x < z^{(1)}$, $g'_{1,n}(x) = \theta g'_{1,n-1}(x)F(x)$. And it is easy to show the remaining properties of this lemma. ■

APPENDIX B: PROOF OF LEMMA 4.2

LEMMA 4.2: *Suppose resources are homogeneous, with $\mathbf{p}_m = (1, \ldots, 1)$. Then the optimal assignment rule is $A^*(\mathbf{x}_m, \mathbf{p}_m, n) = x_m$. That is, one should always accept the job of greatest value.*

PROOF: Suppose that in state $(\mathbf{x}_m, n)$ policy π specifies that job i be assigned, but job m be not assigned, where $x_i < x_m$. Consider a new policy π', which is identical to π except that it interchanges any assignments of job i and m. That is, π' starts by assigning job m, and job i is not assigned. Thereafter π' proceeds as if job i had been assigned, and at a subsequent point (if any) when π would have assigned job m, it now assigns job i. The difference in profits obtained by π' and π is roughly $(x_m + \theta^k) - (x_i + \theta^k x_m)$, for some $k \geq 1$, possibly $k = \infty$. Since this is positive, π' is better than π. ■

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

APPENDIX C: PROOF OF LEMMA 4.3

LEMMA 4.3: *The optimal value function $V_{m,n}(\mathbf{x}_m, \mathbf{p}_m)$ is an increasing and convex function of $\mathbf{x}_m$. Moreover, the right-hand derivative, $\partial V_{m,n}(\mathbf{x}_m, \mathbf{p}_m)/\partial x_i$, is no greater than p_m. In the homogeneous case of $\mathbf{p}_m = (1, \ldots, 1)$, this derivative is no greater than 1 and can be interpreted as the probability that job i is assigned by an optimal policy.*

PROOF: $V_{m,n}(\mathbf{x}_m, \mathbf{p}_m)$ being increasing and convex can be easily proved, by induction on n, by use of the dynamic programming equation:

$$V_{m,n}(\mathbf{x}_m, \mathbf{p}_m) = \max\left\{\theta E\Big[V_{m,n-1}\big(U(\mathbf{x}_m, X)\big)\Big], x_m + V_{m-1,n-1}(\mathbf{x}_{m-1})\right\}.$$

However, there is an alternative proof that is more insightful. Observe that we may apply policy $\pi \in \Pi_{\mathrm{P}}$ to a problem with initial state $(\mathbf{x}_m, \mathbf{p}_m, n)$, we might have applied if the initial state had been $(\mathbf{x}'_m, \mathbf{p}_m, n)$. Under this fixed policy π, the expected profit will be linear and strictly increasing in $\mathbf{x}_m$. The maximum over all such linear and increasing functions (as generated by all possible choices of $\mathbf{x}'_m$ and policies we might apply) must be a strictly increasing and convex function of $\mathbf{x}_m$. This set includes the policy that is optimal for the starting state $(\mathbf{x}_m, \mathbf{p}_m, n)$. So taking this maximum constructs the function $V_{m,n}(\mathbf{x}_m)$, which is therefore strictly increasing and convex.

The above argument also illustrates that $\partial V_{m,n}(\mathbf{x}_m, \mathbf{p}_m)/\partial x_i$ can be interpreted as the expected resource value to which job i is eventually assigned (under an optimal policy). This is no greater than p_m. In the homogeneous case of $\mathbf{p}_m = (1, \ldots, 1)$, we can interpret this derivative as the probability that under an optimal policy job i is assigned. ■

APPENDIX D: PROOF OF THEOREM 4.4

THEOREM 4.1: *With finite m and n, $V_{m,n}(0, \ldots, 0, x) \geq V^{\psi^*}_{m,n}(x)$. Specifically, policy π^* strictly outperforms policy ψ^* under the following conditions: (1) $m = n$ and $x \in (0, z^{(m,m)})$ (with $z^{(1,1)} = z^{(1)}$), (2) $n > m = 1$ and $x \in [0, z^{(1)})$, or (3) $n > m \geq 2$. Finally, in the case of infinite n, $z^{(m)} = w^{(m)}$ and $V_m(0, \ldots, 0, x) = V^{\psi^*}_m(x)$.*

PROOF: We first verify the three conditions under which policy π^* strictly outperforms policy ψ^*. With Lemma 4.1 and Theorem 4.3, it is easy to verify Condition (1) for $m = n = 1$. Based on this conclusion and the monotonicity of $g_{1,n}(x)$ in x, Condition (2) can be established through induction.

Next, we show that $V_{m,m}(0, \ldots, 0) = V^{\psi^*}_{m,m}(0)$ and $V_{m,n}(0, \ldots, 0, 0) > V^{\psi^*}_{m,n}(0)$ for $m < n$. For $m = n$, both policy π^* and policy ψ^* immediately accept all jobs arriving in the future, since $z^{(m,m-1)} = w^{(m,m-1)} = 0$. Thus, $V_{m,m}(0, \ldots, 0) = V^{\psi^*}_{m,m}(0)$. For $m < n$,

$$V_{m,n}(0, \ldots, 0) = \theta\left\{\int_0^{w^{(m,n-1)}} V_{m,n-1}(0, \ldots, 0, y)dF(y) + \int_{w^{(m,n-1)}}^1 V_{m,n-1}(0, \ldots, 0, y)dF(y)\right\}$$

$$V^{\psi^*}_{m,n}(0) = \theta\left\{\int_0^{w^{(m,n-1)}} V^{\psi^*}_{m,n-1}(0)dF(y) + \int_{w^{(m,n-1)}}^1 V^{\psi^*}_{m,n-1}(y)dF(y)\right\}.$$

By the above expressions, since $V_{m,m}(0, \ldots, 0, y)$ increases in y (by Theorem 4.2), the conclusion holds for the case of $n = m + 1$. By repeating a similar argument, we conclude that $V_{m,n}(0, \ldots, 0, 0) > V^{\psi^*}_{m,n}(0)$ for $m < n$.

For $x < w^{(m,m)}$, $V_{m,m}(0, \ldots, 0, x) > V_{m,m}(0, \ldots, 0) = V^{\psi^*}_{m,m}(0) = V^{\psi^*}_{m,m}(x)$. For $x \geq w^{(m,m)}$, $V_{m,m}(0, \ldots, 0, x) \geq x + V_{m-1,m}(0, \ldots, 0) > x + V^{\psi^*}_{m-1,m}(0)$. Thus, under Condition (1), policy π^* strictly outperforms policy ψ^*. Condition (3) can be verified similarly.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

Finally, consider the case with infinite n. It suffices to show that $w^{(m)} = z^{(m)}$. The proof is by induction. The case of $m = 1$ is obvious. Suppose $z^{(m-1)} = w^{(m-1)}$. By (10),

$$\begin{aligned} z^{(m)} &= g_{1,1}(z^{(m)}) + \theta \int_{z^{(m)}}^{1} \Big[g_{m-1}(z^{(m)}) - g_{m-1}(y) \Big] dF(y) \\ &= g_{1,1}(z^{(m)}) + \theta \int_{z^{(m-1)}}^{1} \Big[z^{(m-1)} - y \Big] dF(y) \\ &= \theta z^{(m-1)} + g_{1,1}(z^{(m)}) - g_{1,1}(z^{(m-1)}). \end{aligned}$$

By summing $z^{(2)}, \ldots, z^{(m)}$ as defined above, we have

$$\begin{aligned} & z^{(m)} + \sum_{k=2}^{m-1} z^{(k)} = \theta \sum_{k=1}^{m-1} z^{(k)} + g_{1,1}(z^{(m)}) - g_{1,1}(z^{(1)}) \\ \Rightarrow\ & z^{(m)} - g_{1,1}(z^{(m)}) = \theta \sum_{k=1}^{m-1} z^{(k)} - \sum_{k=1}^{m-1} z^{(k)}. \end{aligned}$$

By the induction hypothesis and (15), $z^{(m)} - g_{1,1}(z^{(m)}) = w^{(m)} - g_{1,1}(w^{(m)})$. Since $x - g_{1,1}(x)$ is strictly increasing for $x \in [0, z^{(1)})$, $z^{(m)} = w^{(m)}$. ■

APPENDIX E: PROOF OF THEOREM 5.2

THEOREM 5.2: *Given $\mathbf{x}_m, \mathbf{p}_m$, and n, and $x_{i-1} < z^{(1)} \le x_i$, the optimal decision rule accepts job i, $i+1$, …, m and assigns job i to resource i.*

PROOF: It suffices to show the conclusion for the case when $x_1 \le x_2 \le \cdots < z^{(1)} \le x_m$. First, we show that if job m is valued higher than $z^{(1)}$ and to be immediately accepted, it is optimal to assign it to resource m. Note that the expected profit of assigning resource i to job m is

$$p_i x_m + V_{m-1,n}\big(\mathbf{x}_{m-1}, (p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_m)\big).$$

Suppose that the expected discounted value of the job assigned to resource m is x. Now, consider assigning job m to resource m. After this assignment, if resource i is regarded as having value p_m, the expected discounted value of the job assigned to each resource remains the same, except that for resource i this value is x. Moreover, it should be clear that $x < z^{(1)}$, otherwise one can find a policy that outperforms policy π^* for the case of $m = 1$, which is defined in Theorem 4.1. Thus, $p_i x_m + p_m x < p_i x + p_m x_m$ and it is better to assign job m to resource m and the conclusion follows.

Now, we show that it is optimal to immediately accept job m if it is valued higher than $z^{(1)}$. Clearly, the case of $m = 1$ is straightforward. For $m \ge 2$, we establish the conclusion by induction on n. For $n = 1$, we first consider postponement. If a job valued at y arrives during the postponing period, the system transitions to one of $m+1$ possible states depending on its position after being inserted into the queue, and the expected profit is

$$\begin{aligned} V^{\mathrm{POST}}_{m,1}(\mathbf{x}_m, \mathbf{p}_m) &= \theta \sum_{i=1}^{m-1} p_i \Big[x_i F(x_i) + \int_{x_i}^{x_{i+1}} y dF(y) + x_{i+1} \bar{F}(x_{i+1}) \Big] + p_m V^{\mathrm{POST}}_{1,1}(x_m) \\ &< V^{\mathrm{POST}}_{m-1,1}(\mathbf{x}_{m-1}, \mathbf{p}_{m-1}) + p_m V^{\mathrm{POST}}_{1,1}(x_m), \end{aligned}$$

which is clearly smaller than $V_{m-1,1}(\mathbf{x}_{m-1}, \mathbf{p}_{m-1}) + p_m x_m$, the expected profit of immediately accepting job m and assigning it to resource m.

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

Suppose the conclusion holds for the case with m and $n-1$. By the induction hypothesis, at each of the $m+1$ states after a job arrives, it is optimal to immediately accept job m in the queue and assign it to resource m, that is,

$$\begin{aligned} V_{m,n}^{\mathrm{POST}}(\mathbf{x}_m, \mathbf{p}_m) = \theta\Big\{ & V_{m-1,n-1}\big((x_1,\ldots,x_{m-1}),(p_1,\ldots,p_{m-1})\big)F(x_1) \\ & + \int_{x_1}^{x_2} V_{m-1,n-1}\big((y,\ldots,x_{m-1}),(p_1,\ldots,p_{m-1})\big)dF(y) \\ & + \cdots + \int_{x_{m-1}}^{x_m} V_{m-1,n-1}\big((x_2,\ldots,x_{m-1},y),(p_1,\ldots,p_{m-1})\big)dF(y) \\ & + V_{m-1,n-1}\big((x_2,\ldots,x_{m-1},x_m),(p_1,\ldots,p_{m-1})\big)\bar{F}(x_m)\Big\} + p_m V_{1,1}^{\mathrm{POST}}(x_m) \\ & < V_{m-1,n}^{\mathrm{POST}}(\mathbf{x}_{m-1},\mathbf{p}_{m-1}) + p_m V_{1,1}^{\mathrm{POST}}(x_m), \end{aligned}$$

which is clearly smaller than $V_{m-1,n}(\mathbf{x}_{m-1},\mathbf{p}_{m-1}) + p_m x_m$. The proof is complete. ■

APPENDIX F: PROOF OF THEOREM 5.3

THEOREM 5.3: *For $2 \le m \le n$, $V_{m,n}^{\tilde{\pi}}((0,\ldots,0,x_m),\boldsymbol{p}_m) \ge V_{m,n}^{\psi^*}(x_m,\boldsymbol{p}_m)$ and $v_m^{(i,n)} \ge w^{(i,n)}$ for $i = 1,2,\ldots,m$, where n is finite and the inequality is strict for $m < n$. Also, $v_m^{(m,n)}$ is independent of $\boldsymbol{p}_m$.*

PROOF: It suffices to show that $v_m^{(i,n)} \ge w^{(i,n)}$ and $v_m^{(i,n)}$ is independent of $\mathbf{p}_m$. We establish these results through induction on m for cases with $m \ge 2$. For each step of induction on m, we establish the results through induction on n. Also, it should be noted that

$$w^{(i,n)} = \theta\Big[w^{(i,n-1)}F(w^{(i,n-1)}) + \int_{w^{(i,n-1)}}^{w^{(i-1,n-1)}} ydF(y) + w^{(i-1,n-1)}\bar{F}(w^{(i-1,n-1)})\Big].$$

This is indeed one way of establishing (15), which can be found in the literature (e.g., Corollary 1 of Derman, Lieberman, & Ross [6]).

Consider the case of $m = 2$, in which $v_1^{(1,n)} = g_{1,n}(0)$. It suffices to verify that $v_2^{(i,n)} \ge w^{(i,n)}$, $i = 1,2$. First, consider $n = 2$. Clearly, both of the jobs yet to arrive will be accepted. Since $v_1^{(1,1)} = g_{1,1}(0) = w^{(1,1)}$, there is no difference between $\tilde{\pi}$ and ψ^*, and thus $v_2^{(i,2)} = w^{(i,2)}$, for $i = 1,2$. Suppose $v_2^{(i,n-1)} \ge w^{(i,n-1)}$ for $n > 2$. Consider the first arriving job valued at y. If $y > v_1^{(1,n-1)}$, this job should be assigned to resource 2, with the expected discounted value of the job assigned to resource 1 as $v_1^{(1,n-1)}$. If $w^{(2,n-1)} < y \le v_1^{(1,n-1)}$, this job should be assigned to resource 1, leaving $v_1^{(1,n-1)}$ to p_2. If $y \le v_1^{(2,n-1)} = w^{(2,n-1)}$, this job should be kept in the queue. Thus,

$$\begin{aligned} v_2^{(1,n)} &\ge \theta\Big[v_2^{(1,n-1)}F(w^{(2,n-1)}) + \int_{w^{(2,n-1)}}^{v_1^{(1,n-1)}} v_1^{(1,n-1)}dF(y) + \int_{v_1^{(1,n-1)}}^{1} ydF(y)\Big] \\ &> \theta\Big[w^{(1,n-1)}F(w^{(1,n-1)}) + \int_{w^{(1,n-1)}}^{1} ydF(y)\Big] \\ &= w^{(1,n)}, \end{aligned}$$

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press

$$\begin{aligned} v_2^{(2,n)} &\geq \theta\Big[v_2^{(2,n-1)}F(w^{(2,n-1)}) + \int_{w^{(2,n-1)}}^{v_1^{(1,n-1)}} y dF(y) + \int_{v_1^{(1,n-1)}}^{1} v_1^{(1,n-1)} dF(y)\Big] \\ &> \theta\Big[w^{(2,n-1)}F(w^{(2,n-1)}) + \int_{w^{(2,n-1)}}^{w^{(1,n-1)}} y dF(y) + \int_{w^{(1,n-1)}}^{1} w^{(1,n-1)} dF(y)\Big] \\ &= w^{(2,n)}, \end{aligned}$$

where the first and the third inequality follow since in the case of $y \in (0, w^{(2,n-1)}]$, not all jobs in the queue are valued at 0. It is easy to see that $v_2^{(i,n)}$ is independent of $\mathbf{p}_2$.

Suppose $v_{m-1}^{(i,n)} \geq w^{(i,n)}$, $i = 1, 2, \ldots, m-1$. The analysis mimics the case of $m = 2$. First, it is easy to see that $v_m^{(i,m)} = w^{(i,m)}$. Suppose $v_m^{(i,n-1)} \geq w^{(i,n-1)}$. It is not hard to conclude that $v_m^{(i,n)}$ is no less than $w^{(i,n)}$ by the following equation:

$$\begin{aligned} v_m^{(i,n)} &\geq \theta\Big\{v_m^{(i,n-1)}F(w^{(m,n)}) + v_{m-1}^{(i,n-1)}\big[F(v_{m-1}^{(i,n-1)}) - F(w^{(m,n)})\big] \\ &\quad + \int_{v_{m-1}^{(i,n-1)}}^{v_{m-1}^{(i-1,n-1)}} y dF(y) + v_{m-1}^{(i-1,n-1)}\bar{F}(v_{m-1}^{(i-1,n-1)})\Big\} \qquad \textbf{(20)} \\ &> \theta\Big[w^{(i,n-1)}F(w^{(i,n-1)}) + \int_{w^{(i,n-1)}}^{w^{(i-1,n-1)}} y dF(y) + w^{(i-1,n-1)}\bar{F}(w^{(i-1,n-1)})\Big] \\ &= w^{(i,n)}. \end{aligned}$$

Also, $v_m^{(i,n)}$ is independent of $\mathbf{p}_m$. ■

https://doi.org/10.1017/S0269964812000289 Published online by Cambridge University Press