Monotonicity and threshold properties in batch workload models with controlled service times consisting of a sum of geometric phases

Citation for published version (APA):

Document status and date:
Published: 01/01/2001

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.
Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain.
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 16. Jan. 2022
Monotonicity and threshold properties in batch workload models with controlled service times consisting of a sum of geometric phases

G.A.J.F. Brouns
J. van der Wal

/ department of mathematics and computing science
Monotonicity and threshold properties in batch workload models with controlled service times consisting of a sum of geometric phases

Gido A.J.F. Brouns  
ace@win.tue.nl  

Jan van der Wal  
jan.v.d.wal@tue.nl

Eindhoven University of Technology  
Department of Mathematics and Computing Science  
PO Box 513, 5600 MB Eindhoven  
The Netherlands

April 11, 2001

Abstract

We consider two versions of a basic model for on-line decision-making in workflow processes. Under certain regularity conditions we establish various monotonicity properties and we show that there exists an optimal threshold policy for the decision to either continue or abort the service of a job. We partially characterize this optimal policy in terms of the model parameters.

Key words: workload models, termination control, optimal threshold policies, Markov decision processes.

1 Introduction

Brouns and Van der Wal [2] studied a workload model with exponential interarrival times and exponential service phases. Using uniformization, they showed under certain regularity conditions that for any fixed number of periods the optimal admission and termination control policies are of a threshold type. As a consequence of uniformization, for the finite horizon problem the total length of time was not fixed.

In this paper we consider a somewhat different model, featuring a fixed number of jobs awaiting service and a fixed span of time ahead of us. Time is divided into sections of equal length, termed periods. At the beginning of each period one has to decide whether to continue service of the job currently under service or to abort service and commence service of the next job. Under some regularity condition on the expected direct reward structure and the probabilities of success with respect to the job under service, we first derive some monotonicity results. Using these results, we show that there exists an optimal threshold policy for this type of decision. We provide a partial characterization of the optimal policy.
in terms of the model parameters. Next, we derive the same results for a slightly modified model in which *multi aborts* are allowed.

For a discussion of the position of our model in literature and for a motivation of our research, we refer to Bouwens and van der Wal [2].

## 2 Model description

### 2.1 The queueing system

We study a single server queueing system without arrivals, operating according to the FCFS discipline and possessing a buffer that is infinitely large. The workload of a job is the sum of at most $N \geq 1$ service phases.

At the beginning of the process we are confronted with a single batch of jobs awaiting service. For the treatment of these jobs we are allowed a fixed number of periods of equal duration, e.g., $W$ weeks, $D$ days or $H$ hours. During the course of a period the job currently under service either does or does not complete its current service phase. The probability of completion may depend on the number of service phases that have already been completed for this job.

After each period one has to decide whether to continue service of the job currently under service (decision *continue*) or to abort service and start servicing the next job in the queue (decision *abort*). In a period the job under service collects an expected direct reward $r(k)$, where $k$ is the number of service phases that have already been completed for that job, $0 \leq k \leq N$. All expected rewards are assumed to be finite. A job that has received full service, i.e., that has completed $N$ service phases, will not collect any future rewards. Therefore, $r(N) = 0$. Jobs residing in the queue do not collect any rewards either.

Apart from these rewards there are holding costs for the jobs residing in the system, either waiting to be taken into service or being serviced. We assume these costs are convex in the number of jobs, viz, $h(i) \geq 0$ per period when there are $i$ jobs present, where $h(0) = 0$. Future rewards and (holding) costs are not discounted.

See Figure 1 for a graphical representation of the queueing system just described.

![Figure 1: Queueing system under consideration](image-url)
When terminating the job under service, jobs residing in the queue may not be aborted. Therefore, abort actually stands for abort followed by continue at the same point in time. The relaxation that jobs in the queue may be removed as well is treated in section 10.

2.2 Routing mechanism and service times

A job being serviced is said to ‘reside in node k’ if it has completed k service phases, 0 ≤ k ≤ N. If at the start of a period the job under service resides in node k, 0 ≤ k < N, and, consequently, is in its (k+1)-th service phase, then the probability that this (k+1)-th service phase will not (yet) be completed in the oncoming period is pkk, whereas the probability that the service phase will be completed in this period is ∑N j=k+1 pkj = 1 − pkk, termed the probability of success for node k. The probability of success for node N is zero, i.e., pNN = 1.

For 0 ≤ k < N, the probability of success consists of N − k sub-probabilities pkj, k < j ≤ N. For a job residing in node k (0 ≤ k < N), pkj (k < j ≤ N) indicates the probability that the job will complete the (k + 1)-th service phase in the oncoming period and will subsequently jump to node j, skipping all j − k − 1 nodes in between node k and node j. The service phases associated with these nodes are automatically completed upon such a jump.

This framework can act as a reasonable representation of a class of workflow processes encountered in practice, where after each service completion, depending on the outcome of the service process with respect to the job under consideration, the job is redirected to some downstream service phase. By specifying a sufficient number of service phases (tasks), one can include as many alternative routings and outcomes of process instances as one desires. For more background information we refer to BROUNS [1].

See Figure 2 for a graphical representation of this (Markov) routing mechanism. Note that we only allow forward jumps, i.e., jobs will never return to service phases they have already passed. Clearly, ∑N j=k pkj = 1 for all 0 ≤ k ≤ N.

![Figure 2: Markov feed-forward routing mechanism](image)

From our model description it follows that the number of periods required to complete a service phase of a job is geometrically distributed, with a parameter that may depend on the number of service phases that have already been completed, viz, pkk if k phases completed. Hence, provided service is not aborted, the expected number of periods (i.e., the average time) required to complete the (k + 1)-th service phase, 0 ≤ k < N, is 1/pkk.

For technical reasons we specify that if a job has received maximum service, and, as a consequence, has reached (final) node N, then it can be either maintained in the system...
or terminated. Maintaining means that the job remains in node $N$, occupying the server, until it is (finally) decided to terminate the job after all. Maintaining a job yields a direct reward of zero. This specification is in agreement with $r(N) = 0$ and $p_{NN} = 1$.

2.3 State description, objective and assumptions

The state of the system is described by the tuple $(i, k)$, where $i$ is the number of jobs in the system and $k$ is the node the job under service resides in. If $i = 0$, then $k$ is indefinite. To indicate this, the empty system is denoted by $(0, \cdot)$. The process ends when either time is up or the number of jobs in the system reaches zero.

The objective is to maximize the expected profit over an $n$-period time horizon, where profit is defined as reward minus cost. Before we write down the dynamic programming equations for our model, we make two important assumptions.

**Fundamental Assumption 1**

\{**Non-increasing expected direct rewards**\}

The expected direct reward function $r(k)$ is non-increasing in $k$, $0 \leq k \leq N$.

$r(k)$ is the expected direct reward for a job residing in node $k$. In words, Fundamental Assumption 1 says that the more service phases have been completed for a job, the less rewarding it becomes to continue, from a one-period look-ahead point of view.

Together with $r(N) = 0$, Fundamental Assumption 1 implies $r(k) \geq 0$ for all $0 \leq k \leq N$.

**Fundamental Assumption 2**

\{**No expected overtaking**\}

For all $k$, $0 \leq k < N$,

$$\forall M = k, \ldots, N : \sum_{j=k}^{M} p_{kj} \geq \sum_{j=k+1}^{M} p_{k+1,j}.$$  

Fundamental Assumption 2 says that from a statistical point of view, a job residing in node $k$ will not overtake a job residing in node $k+1$. This assumption is equivalent to the assumption that for all $k$, $0 \leq k < N$,

$$\forall M = k, \ldots, N : \sum_{j=M+1}^{N} p_{kj} \leq \sum_{j=M+1}^{N} p_{k+1,j}.$$  

**Remark 1**

*Given that the job under service currently resides in node $k$, for some $0 \leq k < N$, let discrete random variable $X_k$ denote the node this job will reside in during the next period, provided service is not aborted. Then Fundamental Assumption 2 states that $X_k \leq_{st} X_{k+1}$, i.e., $X_{k+1}$ is stochastically larger than $X_k$.*

**Remark 2**

*If jobs can only jump from $k$ to $k+1$ for all $0 \leq k < N$, i.e., $p_{k,k+1} = 1 - p_{kk}$ for all $0 \leq k < N$, then Fundamental Assumption 2 is surely satisfied.*
Remark 3
The situation that each job taken into service collects exactly one reward, namely \( s(k) \) if the ultimate number of completed service phases for that job reaches \( k \), can be modelled by defining \( r(k) := \sum_{j=1}^{N} p_{kj}[s(j) - s(k)] \) for all \( k \). So, for a job residing in node \( k \), \( r(k) \) is the expected one-period increase in the overall reward for that job.

Remark 4
Assume jobs are served either successfully (yielding a reward of 1) or unsuccessfully (yielding a reward of 0). For any job, let discrete random variable \( X \) denote the number of periods of time required until success. Here, for all \( k \geq 0 \), \( F(k) := \mathbb{P}(X \leq k) \), \( p_k := \mathbb{P}(X = k) \) and
\[
q_k := \mathbb{P}(X = k|X > k - 1) = \frac{p_k}{1 - F(k - 1)} = \frac{p_k}{1 - \sum_{j=0}^{k-1} p_j}.
\]
The function \( q_k \) represents the success rate. Assume this success rate is non-increasing in \( k \), i.e., the longer we work on a job, the less likely it becomes we will ultimately have success with respect to this job. This 'Decreasing Success Rate' model is a special case of our model, with \( r(k) = q_k \) and \( p_{k,k+1} = 1 \) for all \( k \).

2.4 Mathematical model formulation

Let \( V_n(i, k) \) denote the maximum expected \( n \)-period (additional) profit corresponding to current state \((i, k)\), just before the next decision to continue or abort. At the start of the process, the system is assumed to be in such a state.

Let \( V_n(i, k, \phi) \) denote the expected \( n \)-period (additional) profit corresponding to current state \((i, k)\), given that the next decision is \( \phi \), where \( \phi \in \{\text{continue, abort}\} \), and that thereafter the optimal policy is used. Let \( \pi^* \) denote the optimal decision, so \( V_n(i, k) = V_n(i, k, \pi^*) \).

Then our model is defined by the following dynamic programming equations. To save space, we will usually write \( \text{ab} \) for \( \text{abort} \) and \( \text{co} \) for \( \text{continue} \).

\[
\begin{align*}
V_0(i, k) &= 0 & i \geq 0 \\
&= 0 & 0 \leq k \leq N \\
\text{and, for } n \geq 1, \\
V_n(0, \cdot) &= V_n(0, \cdot, \text{co}) = 0 \\
V_n(i, k) &= \max\{V_n(i, k, \text{co}), V_n(i, k, \text{ab})\} & i \geq 1 \\
&= 0 & 0 \leq k \leq N \\
\text{where} \\
V_n(i, k, \text{co}) &= r(k) + \sum_{j=k}^{N} p_{kj}V_{n-1}(i, j) - h(i) & i \geq 1 \\
&= 0 & 0 \leq k \leq N \\
V_n(i, k, \text{ab}) &= V_n(i - 1, 0, \text{co}) & i \geq 1 \\
&= 0 & 0 \leq k \leq N
\end{align*}
\]
3 Overview of the results

We will prove the following monotonicity and threshold results.

**Proposition 1**
{**V\(_n\)(i, k) NON-INCREASING IN k**}

Let Fundamental Assumptions 1 and 2 be satisfied. Then \(V_n(i, k)\) is non-increasing in \(k\) for all \(n \geq 0\) and \(i \geq 0\), i.e., for all \(0 \leq k < N\),

\[
V_n(i, k) - V_n(i, k + 1) \geq 0.
\]  

(1)

In words, inequality (1) states that residing in upstream nodes is preferred to residing in downstream nodes, irrespective of the remaining number of periods and the number of jobs in the system.

**Proposition 2**
{**MORE MONOTONICITY PROPERTIES**}

Let Fundamental Assumptions 1 and 2 be satisfied. Then, for all \(n \geq 0\),

\[
V_n(i + 1, 0) - V_n(i, 0) \geq V_n(i + 2, k) - V_n(i + 1, k),
\]

(2)

\[
V_n(i + 1, k + 1) - V_n(i, k + 1) \geq V_n(i + 1, k) - V_n(i, k),
\]

(3)

\[
V_{n+1}(i + 1, 0, co) - V_{n+1}(i, 0, co) \geq V_{n+1}(i + 2, k, co) - V_{n+1}(i + 1, k, co),
\]

(4)

\[
V_{n+1}(i + 1, k + 1, co) - V_{n+1}(i, k + 1, co) \geq V_{n+1}(i + 1, k, co) - V_{n+1}(i, k, co),
\]

(5)

where (2) and (4) hold for all \(i \geq 0\) and \(0 \leq k \leq N\), and (3) and (5) for all \(i \geq 1\) and \(0 \leq k < N\).

In words, inequality (2) states that the value of an additional job in state \((i + 1, k)\) does not surpass the value of an additional job in state \((i, 0)\). Analogously, inequality (3) states that the value of an additional job in state \((i, k)\) does not surpass the value of an additional job in state \((i, k + 1)\).

**Remark 5**

Inequalities (2) and (3) imply

\[
V_n(i + 1, k) - V_n(i, k) \geq V_n(i + 2, k) - V_n(i + 1, k)
\]

for all \(n \geq 0\), \(i \geq 1\) and \(0 \leq k \leq N\). I.e., \(V_n(i, k)\) is concave in \(i \geq 1\) for all fixed combinations of \(n \geq 0\) and \(0 \leq k \leq N\).

**Proposition 3**
{**CHARACTERIZATION OF THE OPTIMAL TERMINATION POLICY**}

Let the remaining number of periods be \(n \geq 1\). Then the optimal termination policy can be characterized as follows: if it is optimal to abort the service of a job in state \((i, k)\), then it is optimal as well to abort service in all states \((j, l)\) with \(j \geq i\) and \(l \geq k\).
Figure 3: Characterization of the optimal termination policy

Figure 3 gives a graphical representation of the structure of a typical termination policy. The solid dots represent nodes for which jobs are terminated. The solid polyline captures the termination region.

Note that the termination region is determined by a non-increasing step-function such that for all $i$ there exists a threshold node $k$ of $i$ and for all $k$ there exists a threshold level $i$ of $k$.

**Proposition 4**

**{PARTIAL CHARACTERIZATION OF THE TERMINATION REGION IN TERMS OF THE MODEL PARAMETERS}**

Let the remaining number of periods be $n \geq 1$. In all states $(i,k)$ with either $\{i > n\}$ or $\{1 \leq i \leq n \text{ and } r(k) \leq h(i)\}$, the optimal decision is to abort service.

In practice, Proposition 4 can be used to reduce the state and action spaces and hence the number of dynamic programming equations.

4 **The line of proof**

The main technique to prove the various propositions will be to use induction on the remaining number of periods. In our proofs of Propositions 1 and 2 we will make use of the following lemma.

**Lemma 1**

Let Fundamental Assumption 2 be satisfied. Let $\xi(j)$ be a function which is defined on the points $j = 0, 1, \ldots, N$ and which is non-increasing [non-decreasing] in $j$. Then $\sum_{j=k}^{N} p_{kj} \xi(j)$ is non-increasing [non-decreasing] in $k$, $0 \leq k \leq N$.

**Proof.** Cf Remark 1 and use for example Proposition 9.1.2 of Ross [4].

\qed
5 Proof of Proposition 1 and intermediate threshold result

Throughout this section Fundamental Assumptions 1 and 2 are in effect.

**Proof of Proposition 1.** Inequality (1) holds by definition for \(i = 0\) or \(n = 0\). Assume that for some \(n \geq 0\), inequality (1) holds for all \(i \geq 1\) and \(0 \leq k < N\). This will be our induction hypothesis.

Let \(i \geq 1\) and \(0 \leq k < N\). The next decision prescribed by the (optimal) policy corresponding to \(V_{n+1}(i, k+1)\), is either to continue or to abort the job currently under service. Suppose that abort is optimal. Then

\[
V_{n+1}(i, k) - V_{n+1}(i, k+1) = V_{n+1}(i, k) - V_{n+1}(i, k+1, ab)
\]

\[
\geq V_{n+1}(i, k, ab) - V_{n+1}(i, k+1, ab)
\]

\[
= V_{n+1}(i - 1, 0, co) - V_{n+1}(i - 1, 0, co)
\]

\[
= 0.
\]

Alternatively, suppose that continue is optimal. Then, for \(0 \leq k < N\),

\[
V_{n+1}(i, k) - V_{n+1}(i, k+1) = V_{n+1}(i, k) - V_{n+1}(i, k+1, co)
\]

\[
\geq V_{n+1}(i, k, co) - V_{n+1}(i, k+1, co)
\]

\[
= r(k) + \sum_{j=k}^{N} p_{kj}V_n(i, j) - h(i) -
\]

\[
[r(k+1) + \sum_{j=k+1}^{N} p_{k+1,j}V_n(i, j) - h(i)]
\]

\[
\geq \{r(k)\\text{ non-increasing in } k\}
\]

\[
\sum_{j=k}^{N} p_{kj}V_n(i, j) - \sum_{j=k+1}^{N} p_{k+1,j}V_n(i, j)
\]

\[
\geq \{\text{Fundamental Assumption 2;}
\}

\{\text{induction hypothesis;}
\}

\{\text{Lemma 1 with } \xi(j) = V_n(i, j) \text{ (non-increasing in } j)\}

0.

\[\square\]

**Corollary 1**

Let \(n \geq 1\), \(i \geq 1\) and \(0 \leq k \leq N\). If it is optimal to abort the service of a job in state \((i, k)\), then it is optimal to abort service in all states \((i, l)\) with \(k \leq l \leq N\).

**Proof.** Let \(n \geq 1\) and \(i \geq 1\). Suppose that for some \(0 \leq k \leq N\),

\[
V_n(i, k) = V_n(i, k, ab).
\]
Then, for \( k \leq l \leq N \),
\[
V_n(i, l, ab) = V_n(i - 1, 0, co) = V_n(i, k, ab) = V_n(i, k) \geq \{\text{Proposition 1}\} V_n(i, l).
\]
Since \( V_n(i, l, ab) \leq V_n(i, l) \) by definition, it follows that
\[
V_n(i, l) = V_n(i, l, ab).
\]

6 Proof of Proposition 2 and intermediate threshold result

Throughout this section Fundamental Assumptions 1 and 2 are in effect.

It can easily be verified that inequalities (2) and (3) hold by definition for \( n = 0 \). The proof of Proposition 2 is further organized as follows. Assuming that (2) and (3) hold for some \( n \geq 0 \) (Step 0), we prove that (4) and (5) hold (Step 1). Finally, we prove that (2) and (3) also hold for \( n + 1 \) (Step 2). To establish this, we will make use of two additional lemmas.

**Lemma 2**
Let \( s_m = (i_m, k_m) \), \( m = 1, \ldots, 4 \), and let \( \phi \) and \( \psi \) be authorized decisions, given \( s_m \). Then
\[
V_n(s_1, \phi) - V_n(s_2, \pi^*) \geq V_n(s_3, \pi^*) - V_n(s_4, \psi) \tag{7}
\]
implies
\[
V_n(s_1, \pi^*) - V_n(s_2, \pi^*) \geq V_n(s_3, \pi^*) - V_n(s_4, \pi^*). \tag{8}
\]

**Proof.** Assume (7) for certain decisions \( \phi \) and \( \psi \). Then the result follows immediately from \( V_n(s_1, \pi^*) \geq V_n(s_1, \phi) \) and \( V_n(s_4, \pi^*) \geq V_n(s_4, \psi) \).

\( \Box \)

We will use Lemma 2 in the following way. When distinguishing between all possible combinations of optimal decisions in certain states \( s_2 \) and \( s_3 \), we choose \( \phi \) and \( \psi \) such that (7) holds.

**Lemma 3**
For all \( 0 \leq k \leq N \),
\[
V_{n+1}(1, k) \geq V_n(1, k), \quad n \geq 0, \tag{8}
\]
and
\[
V_{n+1}(1, k, co) \geq V_n(1, k, co), \quad n \geq 1. \tag{9}
\]

**Proof.** First consider (8). For \( 0 \leq k \leq N \), \( V_0(1, k) = 0 \) and \( V_1(1, k) \geq V_1(1, k, ab) = 0 \). Furthermore, \( V_0(0, \cdot) = V_1(0, \cdot) = 0 \). So on \( S_1 := \{0, 1\} \cup \{(1, k)\} \) we have \( V_1 \geq V_0 \). It follows by induction that for all \( n \geq 0 \), \( V_{n+1} \geq V_n \) on \( S_1 \). Note that \( S_1 \) is closed under all policies.

9
Next consider (9). For $0 \leq k \leq N$ and $n \geq 1$,

$$V_{n+1}(1, k, \cdot) - V_n(1, k, \cdot) = \sum_{j=k}^{N} p_{kj} [V_n(1, j) - V_{n-1}(1, j)] \geq \{(8)\} 0.$$

\[\square\]

**Proof of Proposition 2.**

**Step 0.** Since $V_0(i, k) = 0$ for all $i \geq 0$ and $0 \leq k \leq N$, inequalities (2) and (3) hold for $n = 0$.

Assume that for some $n \geq 0$, inequality (2) holds for all $i \geq 0$ and $0 \leq k \leq N$, and (3) for all $i \geq 1$ and $0 \leq k < N$. This will be our induction hypothesis.

**Step 1.** Assuming (2) and (3), we show that (4) and (5) hold for $n + 1$.

We will first prove that (4) holds for $n + 1$. Let $0 \leq k \leq N$. We distinguish between the case that $i = 0$ and the case that $i \geq 1$.

**Case $i = 0$.** We treat this case by giving a sample path argument.

Consider two $(n + 1)$-period process instances of our model, one starting in $(2, k)$, for some $0 \leq k < N$, with the extension that decision continue is chosen (instance $\mathcal{I}_2$), and the other starting in $(1, k)$, also with the extension that decision continue is chosen (instance $\mathcal{I}_1$). We couple all service completion events and, if applicable, the event that the process vanishes, and all decisions (continue versus abort). To be precise, once both processes have carried through the initial continue operation, instance $\mathcal{I}_1$ copies the optimal decisions taken in instance $\mathcal{I}_2$, as long as the single job is in the system. For $\mathcal{I}_2$, let $f^{co}$ denote the policy that prescribes one initial continue operation, followed by the optimal policy after this initial continue operation.

Either $f^{co}$ concerns an abort operation at some point in time, say when $m$ periods remain, $n + 1 > m > 0$, or time hits zero before any abort operations are or can be carried out. In the second case, all decisions in $f^{co}$ are of type continue.

In either case the expected profit for the job in service is equal for $\mathcal{I}_1$ and $\mathcal{I}_2$. If we define

$$\mathcal{P}_{f^{co}}[\text{abort when } m \text{ periods remain}] =: \zeta_m,$$

for $1 \leq m \leq n$, and

$$\sum_{m=1}^{n} \zeta_m =: \zeta,$$

then

$$\mathcal{P}_{f^{co}}[\text{no abort operations}] = 1 - \zeta$$

and

$$V_{n+1}(2, k, \cdot) - V_{n+1}(1, k, \cdot) \leq \sum_{m=1}^{n} \zeta_m [V_m(1, 0, \cdot) - V_m(0, \cdot, \cdot) - (n + 1 - m)[h(2) - h(1)]] +$$

10
\[
(1 - \zeta)[-(n + 1)[h(2) - h(1)] \\
\leq \{h(2) \geq h(1)\} \\
\sum_{m=1}^{n} \zeta_m V_m(1,0,\infty) + (1 - \zeta)[-(n + 1)[h(2) - h(1)] \\
\leq \{Lemma 3\} \\
\zeta V_{n+1}(1,0,\infty) + (1 - \zeta)[-(n + 1)[h(2) - h(1)]] \\
\leq \{r(0) \geq 0; h(i) \text{ convex in } i; h(0) = 0\} \\
\zeta V_{n+1}(1,0,\infty) + (1 - \zeta)[r(0) - h(1)] \\
= \zeta V_{n+1}(1,0,\infty) + (1 - \zeta)[r(0) + \sum_{j=0}^{N} p_{0j} V_n(1,j,ab) - h(1)] \\
\leq \zeta V_{n+1}(1,0,\infty) + (1 - \zeta)[r(0) + \sum_{j=0}^{N} p_{0j} V_n(1,j) - h(1)] \\
= \zeta V_{n+1}(1,0,\infty) + (1 - \zeta) V_{n+1}(1,0,\infty) \\
= V_{n+1}(1,0,\infty).
\]

Case \(i \geq 1\). For \(i \geq 1\) we obtain

\[
V_{n+1}(i + 1,0,\infty) - V_{n+1}(i,0,\infty) = \sum_{j=0}^{N} p_{0j} [V_n(i + 1,j) - V_n(i,j)] - [h(i + 1) - h(i)] \\
\geq \{\text{induction hypothesis; (3)}\} \\
V_n(i + 1,0) - V_n(i,0) - [h(i + 1) - h(i)] \\
\geq \{\text{induction hypothesis; (2)}\} \\
V_n(i + 2,N) - V_n(i + 1,N) - [h(i + 1) - h(i)] \\
\geq \{\text{induction hypothesis; (3); } h(i) \text{ convex in } i\} \\
\sum_{j=k}^{N} p_{kj}[V_n(i + 2,j) - V_n(i + 1,j)] - [h(i + 2) - h(i + 1)] \\
= V_{n+1}(i + 2,k,\infty) - V_{n+1}(i + 1,k,\infty).
\]

This ends our proof of (4) for \(n + 1\).

We will now prove that (5) holds for \(n + 1\). Let \(i \geq 1\) and \(0 \leq k < N\). Then

\[
V_{n+1}(i + 1,k + 1,\infty) - V_{n+1}(i,k + 1,\infty) = \sum_{j=k+1}^{N} p_{kj}[V_n(i + 1,j) - V_n(i,j)] - [h(i + 1) - h(i)] \\
\geq \{\text{Fundamental Assumption 2; Lemma 1 with } \xi(j) = V_n(i + 1,j) - V_n(i,j)\} \\
\sum_{j=k}^{N} p_{kj}[V_n(i + 1,j) - V_n(i,j)] - [h(i + 1) - h(i)] \\
= V_{n+1}(i + 1,k,\infty) - V_{n+1}(i,k,\infty).
\]

This ends our proof of (5) for \(n + 1\).
Step 2. Assuming (2) through (5), we show that (2) and (3) hold for $n + 1$ as well.

We will first prove that (2) holds for $n + 1$. Let $i \geq 0$ and $0 \leq k \leq N$.

The next decision, say $d_1$, prescribed by the (optimal) policy corresponding to $V_{n+1}(i, 0)$, is either to continue or to abort the job under service. Clearly, this also holds for the next decision, say $d_2$, prescribed by the (optimal) policy corresponding to $V_{n+1}(i + 2, k)$. There are at most four joint cases $(d_1, d_2)$. We will show that inequality (2) holds for each case.

The four cases can be presented as follows, where $C$ indicates that continue is optimal and $A$ indicates that continue is not optimal:

\[
\begin{align*}
CC &: V_{n+1}(i, 0, co) > V_{n+1}(i, 0, ab) \land V_{n+1}(i + 2, k, co) > V_{n+1}(i + 2, k, ab), \\
CA &: V_{n+1}(i, 0, co) > V_{n+1}(i, 0, ab) \land V_{n+1}(i + 2, k, co) < V_{n+1}(i + 2, k, ab), \\
AC &: V_{n+1}(i, 0, co) < V_{n+1}(i, 0, ab) \land V_{n+1}(i + 2, k, co) > V_{n+1}(i + 2, k, ab), \\
AA &: V_{n+1}(i, 0, co) < V_{n+1}(i, 0, ab) \land V_{n+1}(i + 2, k, co) < V_{n+1}(i + 2, k, ab).
\end{align*}
\]

The cases $AC$ and $AA$ vanish for $i = 0$, because (by definition) abort is not an option in state $(0, \cdot)$.

Then,
- under $CC$,
  \[
  V_{n+1}(i + 1, 0, co) - V_{n+1}(i, 0) = V_{n+1}(i + 1, 0, co) - V_{n+1}(i, 0, co) \\
  \geq \text{(induction hypothesis; (4))} \\
  V_{n+1}(i + 2, k, co) - V_{n+1}(i + 1, k, co) \\
  = V_{n+1}(i + 2, k) - V_{n+1}(i + 1, k, co); (10)
  \]
- under $CA$,
  \[
  V_{n+1}(i + 1, 0, co) - V_{n+1}(i, 0) = V_{n+1}(i + 1, 0, co) - V_{n+1}(i, 0, co) \\
  = V_{n+1}(i + 2, k, ab) - V_{n+1}(i + 1, k, ab) \\
  = V_{n+1}(i + 2, k) - V_{n+1}(i + 1, k, ab); (11)
  \]
- under $AC$,
  \[
  V_{n+1}(i + 1, 0, ab) - V_{n+1}(i, 0) = V_{n+1}(i + 1, 0, ab) - V_{n+1}(i, 0, ab) \\
  = V_{n+1}(i, 0, co) - V_{n+1}(i - 1, 0, co) \\
  \geq \text{(induction hypothesis; (4))} \\
  V_{n+1}(i + 1, 0, co) - V_{n+1}(i, 0, co) \\
  \geq \text{(induction hypothesis; (4))} \\
  V_{n+1}(i + 1, 0, ab) - V_{n+1}(i, 0, co) \\
  = V_{n+1}(i + 2, k) - V_{n+1}(i + 1, k, co); (12)
  \]
- under $AA$,
  \[
  V_{n+1}(i + 1, 0, ab) - V_{n+1}(i, 0) = V_{n+1}(i + 1, 0, ab) - V_{n+1}(i, 0, ab) \\
  = V_{n+1}(i, 0, co) - V_{n+1}(i - 1, 0, co)
  \]
Finally, apply Lemma 2 to each of the relations (10) through (13) to obtain (2) for \( n + 1 \).

We will now prove that (3) holds for \( n + 1 \). The proof resembles the one of (2) for \( n + 1 \). Let \( i \geq 1 \) and \( 0 \leq k < N \). Again, we distinguish four cases:

- **CC**:

\[
V_{n+1}(i, k + 1, co) - V_{n+1}(i, k, co) \geq V_{n+1}(i, k, ab) - V_{n+1}(i, k, ab)
\]

- **CA**:

\[
V_{n+1}(i, k + 1, ab) \leq V_{n+1}(i, k, ab) - V_{n+1}(i + 1, k, ab)
\]

Then,

- under **CC**,

\[
V_{n+1}(i + 1, k + 1, co) - V_{n+1}(i, k + 1) = V_{n+1}(i + 1, k + 1, ab) - V_{n+1}(i, k + 1, ab)
\]

- under **CA**,

\[
V_{n+1}(i + 1, k + 1, ab) - V_{n+1}(i, k + 1) \geq V_{n+1}(i + 1, k, ab) - V_{n+1}(i, k, ab)
\]

\[
(*) : \{ V_{n+1}(i, k, co) - V_{n+1}(i, k + 1, co) \geq \text{(cf (6);)}
\]

\[
\text{r(k) non-increasing in k; Fundamental Assumption 2; Proposition 1; Lemma 1 with } \xi(j) = V_n(i, j) \text{ (non-increasing in j)}\}
\]

\[
0. \}
\]

- under **AC**,

\[
V_{n+1}(i + 1, k + 1, ab) - V_{n+1}(i, k + 1) = V_{n+1}(i + 1, k + 1, ab) - V_{n+1}(i, k + 1, ab)
\]

\[
\geq \{ \text{induction hypothesis; (4)}\}
\]

\[
V_{n+1}(i + 1, k, co) - V_{n+1}(i, k, co)
\]

\[
= V_{n+1}(i + 1, k) - V_{n+1}(i, k, co).
\]
under $\mathcal{A}$,

$$V_{n+1}(i+1, k+1, ab) - V_{n+1}(i, k+1) = V_{n+1}(i+1, k+1, ab) - V_{n+1}(i, k+1, ab)$$

$$= V_{n+1}(i, 0, co) - V_{n+1}(i - 1, 0, co)$$

$$= V_{n+1}(i+1, k, ab) - V_{n+1}(i, k, ab)$$

$$= V_{n+1}(i+1, k) - V_{n+1}(i, k, ab). \quad (17)$$

Finally, apply Lemma 2 to each of the relations (14) through (17) to obtain (3) for $n + 1$. This ends our proof of (3) for $n + 1$.

This concludes our proof of Proposition 2.

Corollary 2

Let $n \geq 1$, $i \geq 1$ and $0 \leq k \leq N$. If it is optimal to abort the service of a job in state $(i, k)$, then it is optimal to abort service in all states $(j, k)$ with $j \geq i$.

**Proof.** Let $n \geq 1$, $i \geq 1$ and $0 \leq k \leq N$. It suffices to show that

$$V_n(i, k, ab) \geq V_n(i, k, co)$$

implies

$$V_n(i+1, k, ab) \geq V_n(i+1, k, co).$$

Suppose that $V_n(i, k, ab) \geq V_n(i, k, co)$, but $V_n(i+1, k, ab) < V_n(i+1, k, co)$. Then

$$V_n(i, k, ab) = V_n(i - 1, 0, co) \geq V_n(i, k, co),$$

$$-V_n(i+1, k, ab) = -V_n(i, 0, co) > -V_n(i + 1, k, co),$$

so

$$V_n(i, 0, co) - V_n(i - 1, 0, co) < V_n(i + 1, k, co) - V_n(i, k, co),$$

contradicting inequality (4).

7 Proof of Proposition 3

Together, Corollaries 1 and 2 constitute Proposition 3.
8 Proof of Proposition 4

Case \( \{i > n\} \). For \( i > 1 \) and \( 0 \leq k \leq N \),

\[
V_i(i, k, ab) = V_i(i - 1, 0, co) = r(0) - h(i - 1) \\
\geq \{r(k) \text{ non-increasing in } k; h(i) \text{ non-decreasing in } i\} \\
r(k) - h(i) = V_i(i, k, co),
\]

so the statement holds for \( n = 1 \). Assume it holds for some \( n \geq 1 \). This will be our induction hypothesis. Then, for \( i > n + 1 \) and \( 0 \leq k \leq N \),

\[
V_{n+1}(i, k, ab) = V_{n+1}(i - 1, 0, co) = r(0) + \sum_{j=0}^{N} p_{0j} V_n(i - 1, j) - h(i - 1) \\
\geq \{\text{induction hypothesis } (i - 1 > n); \} \\
r(k) \text{ non-increasing in } k; h(i) \text{ non-decreasing in } i\} \\
r(k) + V_n(i - 1, 0, ab) - h(i) \\
\geq \{\text{induction hypothesis } (i - 1 > n)\} \\
r(k) + V_n(i - 1, 0, co) - h(i) \\
= r(k) + \sum_{j=k}^{N} p_{kj} V_n(i, j, ab) - h(i) \\
= \{\text{induction hypothesis } (i > n)\} \\
r(k) + \sum_{j=k}^{N} p_{kj} V_n(i, j) - h(i) = V_{n+1}(i, k, co).
\]

Case \( \{1 \leq i \leq n \text{ and } r(k) \leq h(i)\} \). Let \( 0 \leq k \leq N \) and assume \( r(k) \leq h(1) \). Then

\[
V_n(1, k, ab) = V_1(0, \cdot) = 0 \geq r(k) - h(1) = V_1(1, k, co),
\]

so the statement holds for \( n = 1 \). Assume it holds for some \( n \geq 1 \). This will be our induction hypothesis.

Let \( 1 \leq i \leq n + 1 \), \( 0 \leq k \leq N \) and assume \( r(k) \leq h(i) \). First, we note that

\[
V_{n+1}(i, k, co) - V_n(i, k, co) \geq \min\{r(0) - h(i - 1), 0\}. \tag{18}
\]

This can be explained using coupling: consider an \((n + 1)\)-period process instance and an \(n\)-period process instance, both starting in \((i, k)\) and both with the extension that the initial decision is to continue service. After this initial decision, the \((n + 1)\)-period instance copies the (optimal) decisions taken in the \(n\)-period instance until time hits zero in the \(n\)-period instance. After a total of \(n\) periods either both systems are empty and the difference in profit is 0, or the \((n + 1)\)-period instance has 1 period and at least 1 and at most \(i\) jobs left, so that \texttt{abort} followed by \texttt{continue} results in a difference in profit of at least \(\min\{r(0) - h(i - 1), 0\}\).
Now,

\[ V_{n+1}(i, k, ab) - V_{n+1}(i, k, co) = V_{n+1}(i - 1, 0, co) - \left[ r(k) + \sum_{j=k}^{N} p_{kj} V_{n}(i, j) - h(i) \right] \]

\[ = \{\text{induction hypothesis and } r(k) \text{ non-increasing in } k \text{ if } i < n + 1; \]

\[ \text{case } \{i > n \text{ if } i = n + 1\} \]

\[ V_{n+1}(i - 1, 0, co) - \left[ r(k) + \sum_{j=k}^{N} p_{kj} V_{n}(i, j, ab) - h(i) \right] \]

\[ = \{h(i) - r(k) \geq 0\} \]

\[ \begin{align*}
\quad & \leq \begin{cases}
\{18\} & \quad \min\{r(0) - h(i - 2), 0\} - [r(k) - h(i)] \\
\quad & \quad \min\{r(0) - r(k) + h(i) - h(i - 2), h(i) - r(k)\} \\
\quad & \quad \geq \{r(k) \text{ non-increasing in } k; h(i) \text{ non-decreasing in } i\} \quad 0.
\end{cases}
\end{align*} \]

Note that we did not use Fundamental Assumption 2 in the proof of Proposition 4. Ergo, the proposition does not require this assumption to hold.

9 Counterexamples

If Fundamental Assumption 1 is not obeyed, then inequalities (1) and (4), i.e., the inequalities called upon in the proofs of Corollaries 1 and 2, respectively, need not hold. This can easily be demonstrated by means of elementary counterexamples.

Furthermore, the following trivial counterexample shows that if Fundamental Assumption 1 is indeed obeyed, but Fundamental Assumption 2 is not, then inequality (1) need not hold either.

Counterexample Let \( N = 6 \) and let the rewards be defined as \( r(k) = 12 - 2k, 0 \leq k \leq 6 \). Furthermore, let \( p_{01} \approx 0, p_{02} \approx 1, p_{16} = 1 \) and \( p_{k,k+1} = 1 \) for \( 2 \leq k \leq 5 \).

Let \( i = 1, n = 4 \) and \( h = 0 \), i.e., there is one job in the system and we may serve this job for four periods of time at no cost. It can easily be verified that \( V_{4}(1, 1) = 10 \), whereas \( V_{4}(1, 2) = 20 \). It can be concluded that inequality (1) is violated.

The same instance (with \( n = 2 \)) can be used to show that inequalities (3) and (5) may also be violated in this case, observing that \( V_{2}(2, 1) - V_{2}(1, 1) = 12 > 6 = V_{2}(2, 2) - V_{2}(1, 2) \) and \( V_{2}(2, 1, co) - V_{2}(1, 1, co) = 12 > 6 = V_{2}(2, 2, co) - V_{2}(1, 2, co) \).

Nevertheless, we make the following conjecture.

Conjecture 1 If Fundamental Assumption 1 is obeyed, but Fundamental Assumption 2 is not, then inequality (4) still holds and hence Corollary 2 still holds.
10 Multi aborts

In our model we prescribe that if one decides to abort the service of a job, one immediately starts servicing the next job in the queue, provided the queue is not empty. In practice, this can model the situation that each job in the system has to receive at least some attention (spends at least one period of time at the server), before it may be passed on. This could, for example, concern an administrative task that is associated with each job.

However, if there is no such mandatory initial task, then it will be natural to allow for multi aborts. This means that at a decision moment, one may abort the service of the job currently under service and, at the same time, remove an arbitrary number of jobs from the queue. If the queue ends up empty after this multi abort operation, the process ends. Otherwise, the foremost of the remaining jobs in the queue enters service immediately.

The only dynamic programming equations that change, are the ones for $V_n(i, k, ab)$, which become

$$V_n(i, k, ab) = V_n(i - 1, 0),$$

defined for all $n \geq 1$, $i \geq 1$ and $0 \leq k \leq N$. Consequently,

$$V_n(i, 0) \geq V_n(i - 1, 0)$$

(19)

for all $n \geq 0$ and $i \geq 1$.

In the remainder of this paper, we refer to the situation that multi aborts are allowed as the multi abort case.

**Proposition 5**

*Proposition 1 also holds in the multi abort case.*

**Proof.** In the proof of Proposition 1, replace the passage

$$V_{n+1}(i, k, ab) - V_{n+1}(i, k + 1, ab) = V_{n+1}(i - 1, 0, co) - V_{n+1}(i - 1, 0, co)$$

by

$$V_{n+1}(i, k, ab) - V_{n+1}(i, k + 1, ab) = V_{n+1}(i - 1, 0) - V_{n+1}(i - 1, 0).$$

□

**Proposition 6**

*Inequalities (2) and (3) (from Proposition 2) also hold in the multi abort case.*

**Proof.** Since $V_0(i, k) = 0$ for all $i \geq 0$ and $0 \leq k \leq N$, inequalities (2) and (3) hold for $n = 0$. Assume that for some $n \geq 0$, inequality (2) holds for all $i \geq 0$ and $0 \leq k \leq N$, and (3) for all $i \geq 1$ and $0 \leq k < N$. This will be our induction hypothesis. Under the induction hypothesis we show that (2) and (3) hold for $n + 1$ as well (*Step ii* and *Step iii*, respectively).

**Step ii.** We prove that (2) holds for $n + 1$. Let $i \geq 0$ and $0 \leq k \leq N$. Consider the cases $CC$, $CA$, $AC$ and $AA$ as defined in the first part of Step 2 in the proof of Proposition 2. In our
analysis of case \(\mathcal{AC}\) we also make use of Lemma 3. Note that this lemma holds in the multi abort case (the proof remains exactly the same).

Then,
- under \(CC\),
\[
V_{n+1}(i + 1, 0, \infty) - V_{n+1}(i, 0) = V_{n+1}(i + 1, 0, \infty) - V_{n+1}(i, 0, \infty)
\]
\[
= \sum_{j=0}^{N} p_{0j}[V_n(i + 1, j) - V_n(i, j)] - [h(i + 1) - h(i)]
\]
\[
\geq \{h(i) \text{ convex in } i\}
\]
\[
\sum_{j=0}^{N} p_{0j}[V_n(i + 1, j) - V_n(i, j)] - [h(i + 2) - h(i + 1)]
\]
\[
\geq \{\text{induction hypothesis; (3)}\}
\]
\[
\sum_{j=0}^{N} p_{0j}[V_n(i + 1, 0) - V_n(i, 0)] - [h(i + 2) - h(i + 1)]
\]
\[
= V_{n+1}(i + 1, k, \infty) - V_{n+1}(i + 1, k, \infty); \quad (20)
\]
- under \(CA\) or \(AA\) (case \(AA\) only for \(i > 0\)),
\[
V_{n+1}(i + 1, 0) - V_{n+1}(i, 0) = V_{n+1}(i + 2, k, \infty) - V_{n+1}(i + 1, k, \infty)
\]
\[
\geq V_{n+1}(i + 2, k, \infty) - V_{n+1}(i + 1, k)
\]
\[
= V_{n+1}(i + 2, k) - V_{n+1}(i + 1, k); \quad (21)
\]
- under \(AC\) (\(i > 0\)),
\[
V_{n+1}(i + 1, 0) - V_{n+1}(i, 0)
\]
\[
= V_{n+1}(i + 1, 0) - V_{n+1}(0, \infty)
\]
\[
= V_{n+1}(i + 1, 0) - \max_{0 \leq z \leq \infty} V_{n+1}(z, 0, \infty)
\]
\[
= \{\text{let } z^* \text{ denote the optimal value of } z\}
\]
\[
V_{n+1}(i + 1, 0) - V_{n+1}(z^*, 0, \infty)
\]
\[
\geq \{(19); z^* + 1 \leq i\}
\]
\[
V_{n+1}(z^* + 1, 0) - V_{n+1}(z^*, 0, \infty)
\]
Finally, apply Lemma 2 to the relations (20) and (22) to obtain (2) for \( n + 1 \). Note that we do not need Lemma 2 with respect to relation (21). This ends our proof of (2) for \( n + 1 \).

**Step iii.** We prove that (3) holds for \( n + 1 \). The proof resembles the one of (2) for \( n + 1 \).

Let \( i \geq 1 \) and \( 0 \leq k < N \). Consider the cases CC, CA, AC and AA as defined in the second part of Step 2 in the proof of Proposition 2.

Then,

- under CC,

\[
V_{n+1}(i + 1, k + 1, \omega) - V_{n+1}(i, k + 1) = V_{n+1}(i + 1, k + 1, \omega) - V_{n+1}(i, k + 1, \omega)
\]

\[
= \sum_{j=k}^{N} p_{k,j}[V_n(i + 1, j) - V_n(i, j)] - [h(i + 1) - h(i)]
\]

\[
\geq \{\text{Fundamental Assumption 2; induction hypothesis; (3); Lemma 1 with } \xi(j) = V_n(i + 1, j) - V_n(i, j) \text{ (non-decreasing in } j)\}
\]

\[
= V_{n+1}(i + 1, k + 1, \omega) - V_{n+1}(i, k + 1, \omega);
\]

- under CA,

\[
V_{n+1}(i + 1, k + 1, \omega) - V_{n+1}(i, k + 1) \geq \{\text{Cf corresponding case CA in the proof of Proposition 2,}\}
\]
but replace $V_{n+1}(i, 0, \text{co})$ by $V_{n+1}(i, 0)$ (twice)}
\[ V_{n+1}(i + 1, k) - V_{n+1}(i, k, \text{co}); \]  
(24)

- under $\mathcal{AC}$ or $\mathcal{AA}$,
\[ V_{n+1}(i + 1, k + 1, \text{ab}) - V_{n+1}(i, k + 1) = V_{n+1}(i + 1, k + 1, \text{ab}) - V_{n+1}(i, k + 1, \text{ab}) \]
\[ = V_{n+1}(i, 0) - V_{n+1}(i - 1, 0) \]
\[ \geq \{\text{induction hypothesis; Step ii; (2)}\} \]
\[ V_{n+1}(i + 1, k) - V_{n+1}(i, k). \]  
(25)

Finally, apply Lemma 2 to each of the relations (23), (24) and (25) to obtain (3) for $n + 1$. This ends our proof of (3) for $n + 1$.

This concludes our proof of Proposition 6.

\[ \square \]

Proposition 7

Corollaries 1 and 2 also hold in the multi abort case and hence Proposition 3 holds in the multi abort case.

Proof. In the proof of Corollary 1, replace $V_n(i, l, \text{ab}) = V_n(i - 1, 0, \text{co})$ by $V_n(i, l, \text{ab}) = V_n(i - 1, 0) \geq V_n(i - 1, 0, \text{co})$ to obtain Corollary 1 for the multi abort case.

To obtain Corollary 2 for the multi abort case, follow the proof of Corollary 2 up to the assumption that $V_n(i, k, \text{ab}) \geq V_n(i, k, \text{co})$, but $V_n(i + 1, k, \text{ab}) < V_n(i + 1, k, \text{co})$. Then $V_n(i, k) = V_n(i - 1, 0)$ and $V_n(i + 1, k) > V_n(i, 0)$, so $V_n(i, 0) - V_n(i - 1, 0) < V_n(i + 1, k) - V_n(i, k)$, contradicting inequality (2).

\[ \square \]

Proposition 8

Proposition 4 also holds in the multi abort case.

Proof.

Case $\{ i > n \}$. In the proof of Proposition 4, case $\{ i > n \}$, replace $V_1(i, k, \text{ab}) = V_1(i - 1, 0, \text{co})$ by $V_1(i, k, \text{ab}) = V_1(i - 1, 0) \geq V_1(i - 1, 0, \text{co})$, replace $V_{n+1}(i, k, \text{ab}) = V_{n+1}(i - 1, 0, \text{co})$ by $V_{n+1}(i, k, \text{ab}) = V_{n+1}(i - 1, 0) \geq V_{n+1}(i - 1, 0, \text{co})$, and replace
\[ r(k) + V_n(i - 1, 0, \text{ab}) - h(i) \geq \{\text{induction hypothesis (} i - 1 > n \text{)}\} \]
\[ r(k) + V_n(i - 1, 0, \text{co}) - h(i) \]
by
\[ r(k) + V_n(i - 1, 0, \text{ab}) - h(i) = \{\text{induction hypothesis (} i - 1 > n \text{)}\} \]
\[ r(k) + V_n(i - 1, 0) - h(i). \]

Case $\{ 1 \leq i \leq n \text{ and } r(k) \leq h(i) \}$. Follow the proof of Proposition 4, case $\{ 1 \leq i \leq n \text{ and } r(k) \leq h(i) \}$, up to (18). Instead of using (18), we note that
\[ V_{n+1}(i, k) - V_n(i, k) \geq 0, \]  
(26)
which can be explained using coupling (let the \((n+1)\)-period instance copy the (optimal) decisions taken in the \(n\)-period instance for the next \(n\) periods and terminate all remaining jobs afterwards).

Then

\[
V_{n+1}(i, k, ab) - V_{n+1}(i, k, co) = V_{n+1}(i-1, 0) - [r(k) + \sum_{j=k}^{N} p_{kj} V_{n}(i, j) - h(i)]
\]

\[
= \{\text{induction hypothesis and } r(k) \text{ non-increasing in } k \text{ if } i < n + 1; \}
\]

\[
\text{case } \{i > n\} \text{ if } i = n + 1 \}
\]

\[
V_{n+1}(i-1, 0) - [r(k) + \sum_{j=k}^{N} p_{kj} V_{n}(i, j, ab) - h(i)]
\]

\[
= V_{n+1}(i-1, 0) - [r(k) + V_{n}(i-1, 0) - h(i)]
\]

\[
\geq \{\text{(26); trivial for } i = 1\} \quad h(i) - r(k) \geq 0.
\]

\[
\]

11 Conclusions

We have considered a single server batch workload model with controlled service times consisting of a sum of geometric phases. We have dealt with the periodic decision to either continue or abort the service of a job. The outcome of this decision is a trade-off between expected future direct rewards and holding costs. During a period, the job under service collects an expected direct reward which depends on the number of phases that have already been completed for the job. Besides rewards there are holding costs for the jobs residing in the system. Under two important assumptions, one concerning the expected direct reward function and the other concerning the transition probabilities, we have established various monotonicity properties of the maximum expected \(n\)-period profit function. Using these results, we have shown that the optimal strategy for the continue/abort decision is characterized by a threshold policy, viz, quit serving if there is too much work waiting and if the job under service has already passed a sufficient number of service phases.

The research is motivated by issues arising in workflow control, which captures the on-line decision making process in workflow management. For in-depth discussions of workflow management, we refer to LAWRENCE [3]. For an elaborate treatment of some quantitative analytical aspects of workflow management, we refer to BROUNS [1]. Our workload model covers an important aspect of most workflow problems, namely the fact that there is not enough capacity to treat all jobs completely.

Workflow problems encountered in practice are usually far more complicated. Nevertheless, we are convinced that a better understanding of the optimal strategies for basic models, as the one considered in this paper, will help us to calculate very good strategies for more general workflow problems.
References


