The maximization of CP utilization in an exponential CP-Terminal system with different think times and different job sizes
van der Wal, J.

Published: 01/01/1982

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computing Science

Memorandum COSOR 82-14

The maximization of CP utilization in an exponential CP-Terminal system with different think times and different job sizes

by

J. van der Wal, Eindhoven

Eindhoven, the Netherlands

December 1982
THE MAXIMIZATION OF CP UTILIZATION IN AN
EXPONENTIAL CP-TERMINAL SYSTEM WITH DIFFERENT
THINK TIMES AND DIFFERENT JOB SIZES

by

J. van der Wal, Eindhoven

Abstract.
This paper deals with the optimization of CP utilization in a CP-Terminal system with exponential job sizes and exponential think times. It is shown that if preemptions of the resume type are allowed the best static priority rule is to give priority to the fastest thinker and is independent of the expected job size.
1. Introduction

Consider the following closed queueing network consisting of a central processor (CP) and N terminals, \( T_1, T_2, \ldots, T_N \).

The system operates as follows. Each of the terminals produces jobs for the CP. After having produced a request the terminal 'goes to sleep' until the CP has serviced it. Then the terminal starts to 'think' about a next job. The think times of terminal \( T_i \) are exponentially distributed with mean \( 1/\lambda_i \) and its job sizes are exponential with mean \( 1/\mu_i \). All think times and job sizes are independent and not known in advance.

If there is more than 1 job at the CP then it has to be decided which job to serve. Preemptions are allowed and are assumed to be of the resume type. So at the CP one has to decide after each arrival or departure of a job which job to serve next. The problem considered in this paper is: which service order maximizes the CP utilization. Clearly, the larger the CP utilization the more work is done, though not necessarily for each individual terminal. So we are dealing with social optimization rather than individual optimization.
In a previous paper [1] the case of equal think times for the terminals (i.e. \( \lambda_i = \lambda \) for all \( i \)) is treated. It turns out that in that case the utilization of the CP is not influenced at all by the order in which the jobs are served. Here it will be shown that if the think times are different, CP utilization is maximized by giving priority to the jobs of the faster thinking terminals.

The problem is attacked as follows. First observe that the expected duration of the CP's idle periods is independent of the scheduling (idle periods are exponentially distributed with mean \( \frac{1}{(\lambda_1 + \lambda_2 + \ldots + \lambda_N)} \)). So it suffices to concentrate on the busy period of the CP. Therefore the problem of maximizing CP utilization will be reformulated as a (semi-) Markov decision process concerning the maximization of the expected busy period duration. The notations and much of the arguments used in the proofs stem from this area.

Since all service and think times are exponential it is sufficient to observe and control the system at epochs upon which the state of the system changes. This leads us to a semi-Markov decision process with decisions at the arrival and departure instants.

The state space of the system is the set \( S \) of all nonempty subsets \( \{1, 2, \ldots, N\} \), i.e. state \( A \in S \) corresponds to the situation that the terminals \( T_i \) with \( i \in A \) have delivered a job to the CP and are asleep now, whereas the terminals \( T_i \) with \( i \not\in A \) are thinking. (In order to study the busy period it suffices to consider the nonempty subsets; leaving \( S \), or reaching \( \emptyset \), means the end of the busy period.)

A strategy \( f \) is a function on \( S \) such that \( f(A) \in A \) is the index of the terminal to be served in state \( A \). In this paper not all strategies will be consi-
dered. We will restrict the attention to the subset of strategies which correspond to an ordering of the terminals. Such a so-called ordering strategy is characterized by a permutation \( \pi \) of the numbers \( 1, 2, \ldots, N \). Notation \( \pi = (\pi(1), \pi(2), \ldots, \pi(N)) \). The interpretation is as follows: the terminal with highest priority is \( T_{\pi(1)} \), the next highest priority is for \( T_{\pi(2)} \), etc. So according to \( \pi \) the job to be served in state \( A \) is \( \pi(k) \) if \( k = \min\{k \mid \pi(k) \in A\} \) (by job \( k \) the job from \( T_k \) is meant). In the sequel the ordering will be identified by the corresponding permutation. The set of all permutations of \( 1, \ldots, N \) is denoted by \( \Pi \).

Finally define \( v(A, \pi) \) as the expected remaining busy period duration if the system is now in state \( A \) and ordering \( \pi \) is used. The function \( v(\cdot, \pi) \) satisfies for all \( A \in S \)

\[
(1) \quad v(A, \pi) = \frac{1}{\mu_{\pi(A)}} + \sum_{j \notin A} \lambda_j v(A \setminus \{\pi(A)\}, \pi) + \sum_{j \notin A} \lambda_j v(A \cup \{j\}, \pi),
\]

where \( \pi(A) \) denotes the job to be served in state \( A \) according to \( \pi \).

In this paper the following result will be shown.

**MAIN THEOREM.** Let \( \pi^* = (i_1, i_2, \ldots, i_N) \) be the (an) ordering with

\[
\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N,
\]

then for all \( A \in S \)

\[
v(A, \pi^*) = \max_{\pi \in \Pi} v(A, \pi).
\]

The remainder of the paper is organized as follows. In Section 2 the simplest case \( N = 2 \) is considered. It turns out that if \( \lambda_1 \geq \lambda_2 \) then the ordering \( (i_1, i_2) \) is optimal irrespective of the expected job sizes, thus suggesting
the Main Theorem. In Section 3 a policy iteration type of argument is used to prove the Main Theorem for arbitrary \( N \). The proofs of two lemma's are given in appendices.

2. The case \( N = 2 \)

Let us first consider the simplest version of the problem: the case \( N = 2 \).

For \( N = 2 \) the only state where a decision has to be taken is the state \( \{1,2\} \). So there are only two scheduling strategies, both corresponding to an ordering. (Note that for \( N > 2 \) it is no longer true that any scheduling strategy is an ordering.) For the ordering \( \pi_1 = (1,2) \) the following recursive relations hold (cf. (1)).

\[
\begin{align*}
v(\{1\}, \pi_1) &= \frac{1}{\mu_1 + \lambda_2} \left[ 1 + \lambda_2 v(\{1,2\}, \pi_1) \right] \\
v(\{2\}, \pi_1) &= \frac{1}{\mu_2 + \lambda_1} \left[ 1 + \lambda_1 v(\{1,2\}, \pi_1) \right] \\
v(\{1,2\}, \pi_1) &= \frac{1}{\mu_1} \left[ 1 + \mu_1 v(\{2\}, \pi_1) \right].
\end{align*}
\]

Similarly one has for the ordering \( \pi_2 = (2,1) \)

\[
\begin{align*}
v(\{1\}, \pi_2) &= \frac{1}{\mu_1 + \lambda_2} \left[ 1 + \lambda_2 v(\{1,2\}, \pi_2) \right] \\
v(\{2\}, \pi_2) &= \frac{1}{\mu_2 + \lambda_1} \left[ 1 + \lambda_1 v(\{1,2\}, \pi_2) \right] \\
v(\{1,2\}, \pi_2) &= \frac{1}{\mu_2} \left[ 1 + \mu_2 v(\{1\}, \pi_2) \right].
\end{align*}
\]
Solving the systems (2) and (3) yields

\[
\begin{align*}
\nu((1), \pi_1) &= \frac{1}{\mu_1} + \frac{\lambda_2}{\mu_1 \mu_2} + \frac{\lambda_2 (\lambda_1 - \lambda_2)}{\mu_1 \mu_2 (\lambda_2 + \mu_1)}, \\
\nu((1), \pi_2) &= \frac{1}{\mu_1} + \frac{\lambda_2}{\mu_1 \mu_2}, \\
\nu((2), \pi_1) &= \frac{1}{\mu_2} + \frac{\lambda_1}{\mu_1 \mu_2}, \\
\nu((2), \pi_2) &= \frac{1}{\mu_2} + \frac{\lambda_1 (\lambda_2 - \lambda_1)}{\mu_1 \mu_2 (\lambda_1 + \lambda_2)}, \\
\nu((1, 2), \pi_1) &= \frac{1}{\mu_1} + \frac{1}{\mu_2} + \frac{\lambda_1}{\mu_1 \mu_2}, \\
\nu((1, 2), \pi_2) &= \frac{1}{\mu_1} + \frac{1}{\mu_2} + \frac{\lambda_2}{\mu_1 \mu_2}.
\end{align*}
\]

From this it is easily seen that if \( \lambda_1 > \lambda_2 \) then ordering \( \pi_1 \) is better, i.e. for all A,

\[ \nu(A, \pi_1) > \nu(A, \pi_2). \]

And if \( \lambda_1 = \lambda_2 \) then for all A,

\[ \nu(A, \pi_1) = \nu(A, \pi_2). \]

These results are irrespective of the job size parameters \( \mu_1 \) and \( \mu_2 \).

So for \( N = 2 \) the Main Theorem holds.

3. The general case

In the previous section the Main Theorem has been established for the case \( N = 2 \). In this section the case of arbitrary \( N \) will be treated.
The proof is based on the following result

**Theorem 1.** Let \( \pi_1 \) and \( \pi_2 \) be the two neighbouring orderings \( \pi_1 = (1, 2, \ldots, k - 1, k, k + 1, k + 2, \ldots, N) \) and \( \pi_2 = (1, 2, \ldots, k - 1, k + 1, k, k + 2, \ldots, N) \) then for all \( A \in S \)

(i) if \( \lambda_{k+1} > \lambda_k \) then \( \nu(A, \pi_2) > \nu(A, \pi_1) \)

(ii) if \( \lambda_{k+1} = \lambda_k \) then \( \nu(A, \pi_2) = \nu(A, \pi_1) \).
For notational convenience the theorem has been formulated for the orderings \( \pi_1 \) and \( \pi_2 \) but it will be clear how to extend the result to the case of any two neighbouring orderings \((j_1, \ldots, j_k, j_{k+1}, \ldots, j_N)\) and \((j_1, \ldots, j_{k+1}, j_k, \ldots, j_N)\).

From Theorem 1 it follows that an ordering can be improved by repeatedly interchanging the priorities of two 'neighbouring terminals' until finally the ordering \( \pi^* \) of the Main Theorem is obtained. So, once Theorem 1 is proved the Main Theorem has been established as well.

The proof of Theorem 1 will be given via a sequence of lemmas. Roughly it runs as follows. For the states \( A \) with \( j \notin A \) for \( j = 1, \ldots, k-1 \) and \( k \in A \) and \( k+1 \in A \) we compare for \( \pi_1 \) and \( \pi_2 \) the time until for the first time both terminals \( T_k \) and \( T_{k+1} \) are thinking again. It will be shown that this expected time for \( \pi_2 \) is larger than [equal to] the one for \( \pi_1 \) if \( \lambda_{k+1} > \lambda_k \) (\( \lambda_{k+1} = \lambda_k \)), and that at this time the state under \( \pi_2 \) is stochastically at least as attractive (from the point of maximizing remaining busy period duration) as the one for \( \pi_1 \). Then the proof follows from a rather technical lemma from the area of Markov decision processes.

We will start by giving this lemma but therefore first some notations have to be given.

Let \( S_1 \) be the set of states containing at least one of the elements \( k \) and \( k+1 \) and \( S_2 \) the set of states containing neither \( k \) nor \( k+1 \), so \( S_2 = S \setminus S_1 \).

Now define the stopping time \( \tau \) for the process as follows:

(i) if the initial state of the system lies in \( S_1 \) then \( \tau \) is the time of the first exit from \( S_1 \),

(ii) if the initial state lies in \( S_2 \) then \( \tau \) is the time of the first state change.
Note that the stopping time \( \tau \) depends on the initial state \( A \) and the ordering \( \pi \). Notation \( \tau(A, \pi) \). Clearly \( \tau(A, \pi_1) \) and \( \tau(A, \pi_2) \) are identical for all \( A \in S_2 \), but for \( A \in S_1 \) the distribution functions of \( \tau(A, \pi_1) \) and \( \tau(A, \pi_2) \) may differ.

Further define \( r_\tau(A, \pi) \) as the expected value of \( \tau(A, \pi) \) and \( p_\tau(A, B, \pi) \) as the probability that if the system is in state \( A \) at time 0 and ordering \( \pi \) is used, the system will be in state \( B \) at time \( \tau(A, \pi) \).

Clearly we have

\[
(5) \quad r_\tau(A, \pi) + \sum_B p_\tau(A, B, \pi) v(B, \pi) = v(A, \pi)
\]

(i.e. using \( \pi \) until time \( \tau \) and again thereafter is the same as using \( \pi \) forever).

Now the lemma can be given

**Lemma 1.** (see Wessels [2, Theorem 1.1])

Let \( \pi \) and \( \pi' \) be two orderings. If for all \( A \in S \)

\[
(6) \quad r_\tau(A, \pi') + \sum_B p_\tau(A, B, \pi') v(B, \pi) \geq v(A, \pi)
\]

then for all \( A \in S \)

\[
(7) \quad v(A, \pi') \geq v(A, \pi).
\]

Moreover, if the inequality in (6) is strict for at least one state \( A \) then the inequality in (7) is strict for all states, since all states communicate. If (6) holds with equality for all \( A \), then so does (7).

So, if (6) holds and there is strict inequality for at least one state, then the ordering \( \pi' \) is strictly better than \( \pi \).
Combining (5) and Lemma 1 yields

Theorem 2. Let $\pi_1$ and $\pi_2$ be the two orderings of Theorem 1. If

(8) \[ r_\tau(A, \pi_2) \geq r_\tau(A, \pi_1) \quad \text{for all } A \]

and

(9) \[ \sum_B p_\tau(A, B, \pi_2) v(B, \pi_1) \geq \sum_B p_\tau(A, B, \pi_1) v(B, \pi_1) \quad \text{for all } A, \]

then

(10) \[ v(A, \pi_2) \geq v(A, \pi_1) \quad \text{for all } A. \]

Moreover, if for some state $A$ the inequality in (8) (or (9)) is strict then the inequality in (10) is strict for all $A$. And if (8) and (9) hold with equality for all $A$, then so does (10).

The remainder of this section is devoted to proving that if $\lambda_{k+1} > \lambda_k$ then (8) and (9) hold with strict inequality for at least one state, and that if $\lambda_{k+1} = \lambda_k$ then (8), (9) and hence (10) hold with equality for all $A$. Then Theorem 1 immediately follows from Theorem 2.

As remarked before, $\tau(A, \pi_1)$ and $\tau(A, \pi_2)$ are identical on $S_2$, so (8) and (9) hold with equality for all $A \in S_2$. Therefore let us focus on $S_1$.

The argument is based on the following lemma. The rather complicated proof is postponed to Appendix A.
Lemma 2. Let $F_{\tau(A, \pi)}$ be the distribution function of $\tau(A, \pi)$, i.e.

$$F_{\tau(A, \pi)}(t) = \text{Prob}(\tau(A, \pi) \leq t), \quad t \geq 0.$$ 

Then for all $A \in S_1$

(i) if $\lambda_{k+1} > \lambda_k$ then $F_{\tau(A, \pi_1)}(t) > F_{\tau(A, \pi_2)}(t)$ for all $t > 0$

(ii) if $\lambda_{k+1} = \lambda_k$ then $F_{\tau(A, \pi_1)}(t) = F_{\tau(A, \pi_2)}(t)$ for all $t \geq 0$.

So, if $\lambda_{k+1} > \lambda_k$ then $\tau(A, \pi_2)$ is stochastically larger than $\tau(A, \pi_1)$. And if $\lambda_{k+1} = \lambda_k$ then they are equal.

So for $\lambda_{k+1} = \lambda_k$ the random variables $\tau(A, \pi_1)$ and $\tau(A, \pi_2)$ are identically distributed for all $A \in S$. Hence (8), (9) and (10) hold with equality for all $A \in S$, which proves Theorem 1 (ii).

From now on we can concentrate on the case $\lambda_{k+1} > \lambda_k$. Lemma 2 (i) yields

(11) if $\lambda_{k+1} > \lambda_k$ then $r_{\tau(A, \pi_2)} > r_{\tau(A, \pi_1)}$ for all $A \in S_1$.

This result follows immediately from the following standard lemma by substitution of $h(t) = t$.

Lemma 3. Let $h$ be a nondecreasing [increasing] function on $[0, \infty)$ and let $F$ and $G$ be two distribution functions on $[0, \infty)$ satisfying $F(t) > G(t)$ for all $t > 0$, then

$$\int_0^{\infty} h(t) \, dF(t) > \int_0^{\infty} h(t) \, dG(t).$$
Remains to establish inequality (9). Let $A$ be an initial state in $S_1$ and let $t_j$ be the realization of the remaining think time of terminal $T_j$, $j = k + 2, \ldots, N$, $j \notin A$ at time 0. Define

$$B_t := \{j \in \{k + 2, \ldots, N\} \mid j \in A \text{ or } j \notin A \text{ and } t_j \leq t\}.$$

So $B_t$ is the set of terminals within $\{k + 2, \ldots, N\}$ that have a job at the CP at time $t$ if $t \leq \tau$. Clearly $B_t$ is nondecreasing in $t$.

For these fixed realizations $t_j$ we have

$$\sum_{B} p_{t}(A, B, \pi_{\ell}) v(B, \pi_{\ell}) = \int_{0}^{\infty} v(B_t, \pi_{\ell}) dF_{\tau}(A, \pi_{\ell})(t), \quad \ell = 1, 2.$$

Now (9) is obtained from Lemma 3 if it can be shown that $v(B_t, \pi_{\ell})$ is nondecreasing in $t$. Since $B_t$ is nondecreasing in $t$ this follows immediately from

Lemma 4. Let $\pi$ be any ordering and let $A$ and $B$ be two states with $A \subset B$ then $v(A, \pi) \leq v(B, \pi)$.

The proof of this intuitively appealing result, the more jobs at the CP the longer the remaining busy period, will be given in Appendix B.

Thus (9) holds for all $A \in S_1$. From (11) we have seen that (8) holds with strict inequality for all $A \in S_1$. Further as argued before both (8) and (9) hold with equality for all $A \in S_2$. Since all states in $S$ communicate, i.e. can be reached from each other with positive probability before the end of the busy period, this implies, that if $\lambda_{k+1} > \lambda_k$ then (10) holds with strict inequality for all $A \in S$. Thus the proof of Theorem 1 (i) is complete.

As we have seen before this also establishes the Main Theorem.
4. Conclusion

In the exponential CP-terminal system CP utilization is maximized within the set of orderings by the one which always serves the job of the fastest thinking terminal now asleep. It remains to be shown whether this ordering is also optimal within the set of all scheduling strategies. Tedious computations have shown that also for $N = 3$ an ordering is optimal. So we conjecture it is for all $N$.

Usually one is inclined to serve small jobs first in order to avoid unnecessary waiting. If the faster thinkers produce the smaller jobs then there is no problem but if the slower thinkers produce the smaller jobs then there is a difficulty. Either the small jobs have to wait relatively long or the utilization of the CP decreases. In this respect the question can be raised how seriously the ordering influences the utilization. One slow thinker producing small jobs will not have much influence on the utilization but what happens if there are quite a few of them?
Appendix A. Proof of Lemma 2

Let us start with the case $\lambda_{k+1} > \lambda_k$.

The following three sets of states will play a role in the proof.

- $S_{kk+1} := \{ A \in S | j \notin A, j = 1, \ldots, k-1, k \in A, k+1 \notin A \}$
- $S_k := \{ A \in S | j \notin A, j = 1, \ldots, k-1, k \in A, k+1 \in A \}$
- $S_{k+1} := \{ A \in S | j \notin A, j = 1, \ldots, k-1, k \in A, k+1 \notin A \}$

Clearly on each of the three sets $S_k, S_{k+1}$ and $S_{kk+1}$ the random variables $\tau(A, \pi_1), \tau(A, \pi_2)$, $\tau = 1, 2$, are constant, i.e. independent of whether the terminals $T_{k+2}, \ldots, T_N$ are thinking or asleep. These random variables are denoted by $\tau(k, \pi_1), \tau(k+1, \pi_2)$ and $\tau(k, k+1, \pi_1)$, and their distribution functions by $F_{k, \pi_1}, F_{k+1, \pi_2}$ and $F_{k, k+1, \pi_1}$, respectively.

Note that in all states, except for the states in $S_{kk+1}$, the behaviour of $\pi_1$ and $\pi_2$ is the same. In the proof first $F_{k, k+1, \pi_1}$ and $F_{k, k+1, \pi_2}$ are compared. The argument for initial states $A$ outside $S_{kk+1}$ then easily follows.

The proof is seriously complicated by the fact that the servicing of $k$ and $k+1$ can be interrupted by an arrival of a higher priority job, i.e. a job from one of the terminals $T_1, \ldots, T_{k-1}$. Such interrupts are generated at a rate $\lambda = \lambda_1 + \ldots + \lambda_{k-1}$. By ignoring which terminal generates the interrupt, interrupts can be regarded as 'standard' busy periods of jobs from $T_1, \ldots, T_{k-1}$.

This busy period is denoted by the random variable $\beta$ with distribution function $F_\beta$ and is independent of whether $\pi_1$ or $\pi_2$ is used.
Let us consider $\pi_1$ first. On $S_{kk+1}$ job $k$ (the one from $T_k$) is serviced until one of two things happens

(i) Job $k$ completes service. Then the system moves to $S_{k+1}$ and job $k+1$ is serviced next.

(ii) An interrupt of one of the higher priority terminals $T_1, \ldots, T_{k-1}$ arrives. Then their busy period is serviced. At the end of this busy period the service of $k$ is resumed.

On $S_{k+1}$ job $k+1$ is serviced until one of three things happens

(i) Job $k+1$ completes service. Then time $\tau$ has arrived.

(ii) $T_k$ produces a job. Then the system is back in $S_{kk+1}$.

(iii) A higher priority job arrives. Then this busy period is serviced. At the end of which there are two possibilities. Either $T_k$ has produced a job, then the system moves to $S_{kk+1}$, or $T_k$ has not produced a job and the system returns to $S_{k+1}$.

So when starting in $S_{kk+1}$ and using $\pi_1$ the system cannot reach $S_k$ before $\tau$.

By embedding the process on the sets $S_{k+1}$ and $S_{kk+1}$ the following equations for the distribution functions $F_{k+1,\pi_1}$ and $F_{k,\pi_1}$ are obtained

\begin{align}
F_{k,k+1,\pi_1}(x) &= \text{Prob}(\tau(k,k+1,\pi_1) \leq x) \\
&= \int_0^x (\lambda + \mu_k) e^{-(\lambda+\mu_k)y} \left\{ \frac{\lambda}{\lambda + \mu_k} \text{Prob}(\beta + \tau(k,k+1,\pi_1) \leq x-y) + \frac{\mu_k}{\lambda + \mu_k} \text{Prob}(\tau(k+1,\pi_1) \leq x-y) \right\} dy.
\end{align}

\begin{align}
F_{k+1,\pi_1}(x) &= \int_0^x (\lambda + \mu_{k+1} + \lambda_k) e^{-(\lambda+\mu_{k+1}+\lambda_k)y} \left\{ \frac{\lambda}{\lambda + \mu_{k+1} + \lambda_k} Q(x-y) + \frac{\mu_{k+1} + \lambda_k}{\lambda + \mu_{k+1} + \lambda_k} \text{Prob}(\tau(k,k+1,\pi_1) \leq x-y) \right\} dy.
\end{align}

As
In (A2) \( Q(t) \) denotes the probability that the remaining 'life time' of \( \tau \) is at most \( t \) at the arrival time of a higher priority busy period while servicing a job from \( T_{k+1} \). So \( Q(t) \) satisfies

\[
Q(t) = \int_0^t \left\{ e^{-\lambda_k u} \text{Prob}(\tau(k+1, \pi_1) \leq t-u) + \right. \\
\left. (1 - e^{-\lambda_k u}) \text{Prob}(\tau(k,k+1, \pi_1) \leq t-u) \right\} dF_\beta(u) .
\]

Now define

\[
G_k(t) = \int_0^t e^{-\lambda_k u} dF_\beta(u)
\]

and

\[
H_k(t) = \int_0^t (1 - e^{-\lambda_k u}) dF_\beta(u) .
\]

Then (A3) can be rewritten as

\[
Q(t) = \int_0^t \text{Prob}(\tau(k+1, \pi_1) \leq t-u) dG_k(u) + \\
\left. + \int_0^t \text{Prob}(\tau(k,k+1, \pi_1) \leq t-u) dH_k(u) .
\]

Equations (A1) and (A2), with (A6) substituted into it, can be combined via Laplace-Stieltjes transforms. Denoting the transform of a function \( F(x) \) by \( F^*(w) \), i.e.

\[
F^*(w) = \int_0^\infty e^{-wx} dF(x) ,
\]

we arrive at

\[
P^*_{k,k+1, \pi_1}(w) = \frac{1}{\lambda + \mu_k + \omega} \left[ \lambda F^*_\beta(w) F^*_{k,k+1, \pi_1}(w) + \mu_k F^*_{k+1, \pi_1}(w) \right] .
\]
and

\[
F_{k+1, \pi_1}^*(w) = \frac{1}{\lambda + \mu_{k+1} + \lambda_k + w} \left[ \lambda F_{k+1, \pi_1}^*(w) + \lambda F_{k,k+1, \pi_1}^*(w) H_k^*(w) + \mu_{k+1} + \lambda_k F_{k,k+1, \pi_1}^*(w) \right].
\]

Solving (A7) and (A8) for \(F_{k,k+1, \pi_1}^*(w)\) yields

\[
F_{k,k+1, \pi_1}^*(w) = \mu_k \mu_{k+1} \left[ (\lambda + \mu_{k+1} + \lambda_k + w - \lambda G_k^*(w)) \cdot \right.
\]

\[
\left. \cdot (\lambda + \mu_k + w - \lambda F_S^*(w)) - \mu_k (\lambda H_k^*(w) + \lambda_k) \right]^{-1}.
\]

This completes for the moment the analysis for \(\pi_1\). Now let us continue with the ordering \(\pi_2\).

When \(\pi_2\) is used and the system starts in \(S_{kk+1}\) it can only reach \(S_k\) but never \(S_{k+1}\) before \(\tau\). In \(S_k\), and during the busy periods of \(T_1, \ldots, T_{k-1}\) jobs interrupting the service of job \(k\), terminal \(T_{k+1}\) generates a new job at a rate \(\lambda_{k+1}\). To see what is the effect of \(T_{k+1}\) thinking faster than \(T_k\) (rate \(\lambda_{k+1}\) instead of \(\lambda_k\)) we consider the following modification of the process. A job produced by \(T_{k+1}\) is accepted with probability \(\lambda_k / \lambda_{k+1}\) only. If a job is refused then \(T_{k+1}\) starts to think about another job. This reduces the effective think rate of \(T_{k+1}\) to \(\lambda_k\). These job refusals clearly shorten the time until for the first time both \(T_k\) and \(T_{k+1}\) are thinking again, the stopping time \(\tau\).

To see this, think of the processing of job \(k\) and the \(T_1, \ldots, T_{k-1}\) busy periods interrupting job \(k\) as the 'normal' process and of the \(T_{k+1}\)-jobs as interrupts having as duration the time to move from \(S_{kk+1}\) to \(S_k\).
For the modified process (with $T_{k+1}$ producing jobs at a rate $\lambda_k$ instead of $\lambda_{k+1}$) the time until for the first time both $T_k$ and $T_{k+1}$ are thinking again is denoted by $\sigma(k,\pi_2)$ and $\sigma(k,k+1,\pi_2)$ when the process starts on $S_k$ and $S_{kk+1}$ respectively and $\pi_2$ is used. (So $\sigma$ is actually the stopping time $\tau$ but now for the modified process.) The corresponding distribution functions will be denoted by $\tilde{F}_{k,\pi_2}$ and $\tilde{F}_{k,k+1,\pi_2}$.

From the reasoning above it will be clear that $\tilde{F}_{k,\pi_2}(x) > F_{k,\pi_2}(x)$ for all $x > 0$ and since the time to reach $S_k$ from $S_{kk+1}$ has not been changed by the modification also

\[(A10) \quad \tilde{F}_{k,k+1,\pi_2}(x) > F_{k,k+1,\pi_2}(x) \quad \text{for all} \quad x > 0.
\]

The next step in the proof consists of showing that $\tilde{F}_{k,k+1,\pi_2}(x) = F_{k,k+1,\pi_1}(x)$ for all $x \geq 0$.

Similarly as in the case of $\pi_1$ one derives

\[(A11) \quad \tilde{F}_{k,k+1,\pi_2}(x) = \int_{0}^{x} (\lambda + \mu_{k+1}) e^{-(\lambda + \mu_{k+1})y} \cdot \left\{ \frac{\lambda}{\lambda + \mu_{k+1}} \text{Prob}(\beta + \sigma(k,k+1,\pi_2) \leq x - y) + \frac{\mu_{k+1}}{\lambda + \mu_{k+1}} \text{Prob}(\sigma(k,\pi_2) \leq x - y) \right\} dy.
\]

\[(A12) \quad \tilde{F}_{k,\pi_2}(x) = \int_{0}^{x} (\lambda + \mu_k + \lambda_k) e^{-(\lambda + \mu_k + \lambda_k)y} \cdot \left\{ \frac{\lambda}{\lambda + \mu_k + \lambda_k} R(x - y) + \frac{\mu_k}{\lambda + \mu_k + \lambda_k} + \frac{\lambda_k}{\lambda + \mu_k + \lambda_k} \text{Prob}(\sigma(k,k+1,\pi_2) \leq x - y) \right\} dy,
\]
with

\[ R(t) = \int_0^t \left\{ e^{-\lambda u} \text{Prob}(\sigma(k, \pi_2) \leq t-u) + (1-e^{-\lambda u}) \text{Prob}(\sigma(k, k+1, \pi_2) \leq t-u) \right\} dF_\beta(u) \]

\[ = \int_0^t \text{Prob}(\sigma(k, \pi_2) \leq t-u) dG_k(u) + \int_0^t \text{Prob}(\sigma(k, k+1, \pi_2) \leq t-u) dH_k(u). \]

Transforming (A11) yields

\[ \text{(A14)} \quad \tilde{F}_{k,k+1}^{*, \pi_2}(w) = \frac{1}{\lambda + \mu_{k+1} + \lambda_k + w} \left[ \lambda F_\beta^*(w) \tilde{F}_{k,k+1}^{*, \pi_2}(w) + \mu_k \tilde{F}_{k}^{*, \pi_2}(w) \right]. \]

Substituting (A13) into (A12) and then transforming gives

\[ \text{(A15)} \quad \tilde{F}_{k,k+1}^{*, \pi_2}(w) = \frac{1}{\lambda_k + \mu_{k+1} + \lambda_k + w} \left[ \lambda F_\beta^*(w) \tilde{F}_{k,k+1}^{*, \pi_2}(w) + \lambda \tilde{F}_{k,k+1}^{*, \pi_2}(w) H_k^*(w) + \right. \]

\[ \left. + \mu_{k+1} + \lambda_k \tilde{F}_{k,k+1}^{*, \pi_2}(w) \right]. \]

Now solving (A14) and (A15) for \( \tilde{F}_{k,k+1}^{*, \pi_2}(w) \) results in

\[ \text{(A16)} \quad \tilde{F}_{k,k+1}^{*, \pi_2}(w) = \mu_k \mu_{k+1} \left[ (\lambda + \mu_k + \lambda_k + w - \lambda G_k^*(w))(\lambda + \mu_{k+1} + w - \lambda F_\beta^*(w)) + \right. \]

\[ - \mu_{k+1}(\lambda H_k^*(w) + \lambda_k) \left. \right]^{-1}. \]
Finally it easily follows with $G_k^*(w) + H_k^*(w) = F_\beta^*(w)$ (cf. (B4), (B5)) that the right hand sides in (A9) and (A16) are identical. Hence

$F_{k,k+1,\pi_1}(x) = F_{k,k+1,\pi_2}(x)$ for all $x \geq 0$, and with (A10) also

$F_{k,k+1,\pi_1}(x) > F_{k,k+1,\pi_2}(x)$ for all $x > 0$. This proves Lemma 2 (i) for all initial states $A \in S_{kk+1}$. But then Lemma 2 (i) has to hold for all initial states $A$, since until $S_{kk+1}$ is reached or $\tau$, whatever first, the prescribed behaviour of $\pi_1$ and $\pi_2$ is identical and if $S_{kk+1}$ is reached, which happens with positive probability within any however small amount of time, then from that time onwards $\pi_2$ is strictly better.

The proof for the case $\lambda_{k+1} = \lambda_k$ is simple now. The modification step (the reduction of $\lambda_{k+1}$ to $\lambda_k$) is void now, so $F_{k,k+1,\pi_2} = F_{k,k+1,\pi_2}$ and hence, comparing again (A9) and (A16), also $F_{k,k+1,\pi_1} = F_{k,k+1,\pi_2}$. The argument for the initial states outside $S_{kk+1}$ is similar as in the case $\lambda_{k+1} > \lambda_k$. 

Appendix B. Proof of Lemma 4

Assume $A \subset B$. We will derive a set of recursive relations between the differences $d(D, C) := v(D, \pi) - v(C, \pi)$ for any two states $C$ and $D$ with $C \subset D$. Ultimately the monotonicity of $v$ will follow from the nonnegativity of the function $d$ on the set $V := \{(D, C) | C, D \in S, C \subset D\}$. Let us denote by $\pi(C)$ the index of the terminal to be served in state $C$.

Two cases have to be distinguished. Namely (i) $\pi(D) \notin C$ and (ii) $\pi(D) \in C$.

Case (i). $\pi(D) \notin C$. Then (cf. (i))

\[
(B1) \quad v(D, \pi) = \frac{1}{\mu_{\pi(D)} + \sum_{j \notin D} \lambda_j} \left[ 1 + \mu_{\pi(D)} v(D \setminus \{\pi(D)\}, \pi) + \sum_{j \notin D} \lambda_j v(D \cup \{j\}, \pi) \right].
\]

So,

\[
(B2) \quad d(D, C) = v(D, \pi) - v(C, \pi) >
\]

\[
> \frac{1}{\mu_{\pi(D)} + \sum_{j \notin D} \lambda_j} \left[ \mu_{\pi(D)} (v(D \setminus \{\pi(D)\}, \pi) - v(C, \pi)) + \sum_{j \notin D} \lambda_j (v(D \cup \{j\}, \pi) - v(C, \pi)) \right] =
\]

\[
= \frac{1}{\mu_{\pi(D)} + \sum_{j \notin D} \lambda_j} \left[ \mu_{\pi(D)} d(D \setminus \{\pi(D)\}, C) + \sum_{j \notin D} \lambda_j d(D \cup \{j\}, C) \right].
\]

Case (ii). $\pi(D) \in C$. Observe that, since $\pi$ is an ordering, $\pi(C) = \pi(D)$. Thus we have

\[
(B3) \quad v(C, \pi) = \frac{1}{\mu_{\pi(D)} + \sum_{j \notin C} \lambda_j} \left[ 1 + \mu_{\pi(D)} v(C \setminus \{\pi(D)\}, \pi) + \sum_{j \notin C} \lambda_j v(C \cup \{j\}, \pi) \right].
\]
Further, as one easily verifies, (B1) is equivalent to

$$(B4) \quad v(D,\pi) = \frac{1}{\mu_{\pi(D)}} \left[ 1 + \mu_{\pi(D)} v(D\setminus\{\pi(D)\},\pi) + \sum_{j \notin C} \lambda_j v(D \cup \{j\},\pi) + \sum_{j \in C \setminus D} \lambda_j v(D,\pi) \right].$$

Subtracting (B3) from (B4) yields

$$(B5) \quad d(D,C) = \frac{1}{\mu_{\pi(D)}} \left[ \mu_{\pi(D)} d(D\setminus\{\pi(D)\},C\setminus\{\pi(D)\}) + \sum_{j \notin C} \lambda_j d(D \cup \{j\},C \cup \{j\}) + \sum_{j \in C \setminus D} \lambda_j d(D,C \cup \{j\}) \right].$$

On the set $V$ a transient Markov chain can be introduced. The transition probabilities for the chain follow from (B2) for case (i) and (B5) for case (ii) pairs. For instance, if $\pi(D) \notin C$ then

$$p((D,C)\setminus\{\pi(D)\},C) = \frac{\mu_{\pi(D)}}{\mu_{\pi(D)} + \sum_{j \notin D} \lambda_j}.$$ 

Since if $V$ is left one arrives in $(E,\emptyset)$ with $E \in \{1,2,\ldots,N\}$ and clearly $d(E,\emptyset) \geq 0$, we have by (B2) and (B5) using the function $d$ on $V$ as column-vector

$$d \geq Pd,$$

where $P$ is the transition matrix on $V$. Iterating this we get $d \geq P^n d$ for all $n$ and with $P^n \to 0$ this yields $d \geq 0$. Hence for any two sets $A,B \in S$ with $A \subseteq B$ one has $v(A,\pi) \leq v(B,\pi)$, which completes the proof of Lemma 4.
References
