Scheduling a repairman in a finite source system

Ger Koole[†] *& Martin Vrijenhoek*[*]

[†] *INRIA Sophia Antipolis*

B.P. 93, 06902 Sophia Antipolis

France

[*] *Department of Mathematics and Computer Science*

Leiden University

P.O.Box 9512, 2300 RA Leiden

The Netherlands

The scheduling of a single server in a finite source model is considered. The N customers in the system have different failure and repair rates. Also the costs depend on the customers which are broken down. We give a condition under which the average costs are minimized by a simple list policy, and with a counterexample we show that in the general case no optimal list policy may exist. This motivates us to derive policies which are optimal under low and high traffic conditions. They are again list policies, which behave well numerically.

Keywords: Scheduling, dynamic programming, μc rule, repairman assignment.

Appeared in: ZOR - Mathematical Methods of Operations Research, 44:333–344, 1996.

1. Introduction

In this paper we consider a closed queueing system in which N customers circulate between two stations. One station contains customers which are functioning, the other is a single server queue containing failed customers. Each functioning customer can fail, independently of the other customers, and when that happens it arrives at the queue. The repairman or server can repair the broken customers, though only one at a time. If a customer is repaired, it joins the station with functioning customers (which we call the source).

The system incurs a holding cost c_j for every unit of time that customer j, $1 \leq j \leq N$, is not functioning. The failure and service times are all independent and exponentially distributed with parameters λ_j and μ_j for customer j, respectively.

We are interested in finding the preemptive repair policy that minimizes the total average holding costs. If we choose $c_j = 1$ for every customer j, this results in minimizing the average queue length, a problem which is well studied in the literature. As alternative objective we also consider minimizing the utilization factor of the server.

An important class of policies are the *list policies*. We call a policy a list policy if all customers are ordered (the list) and served according to this order. The list policy that repairs the failed customer with the smallest index available first is called the Smallest Index Policy (SIP). In section 2 we show that the SIP minimizes the holding costs if both $\lambda_1 \leq \cdots \leq \lambda_N$ and $\mu_1 c_1 \geq \cdots \geq \mu_N c_N$. In fact we prove the optimality of the SIP for a larger class of cost function, assuming that $\lambda_1 \leq \cdots \leq \lambda_N$. This class contains also the utilization factor of the server, showing that it is minimized by serving customers in increasing order of λ_j.

Section 3 contains an example showing that, for the holding costs case, there need not be an optimal list policy. This negative result motivates us in section 4 to study policies which are optimal under high or low traffic. There we find that under high traffic the SIP is optimal if $\mu_1 c_1/\lambda_1 \geq \cdots \geq \mu_N c_N/\lambda_N$. Under low traffic the SIP is optimal if $\mu_1 c_1 \geq \cdots \geq \mu_N c_N$.

The direct motivation for this work is the paper by Chakka and Mitrani [3]. They study a model in which tasks arrive to a group of processors. These processors are subject to failure, and there is a single repairman to repair failed processors. Part of their analysis consists of comparing different repairman policies as to maximize the average processing capacity. This is exactly our model, with customers as processors, and c_i the service rate of processor i. (In fact we minimize the weighted inavailability instead of maximizing the weighted availability of the processors.) In [3] different list policies are compared numerically; here we conduct a more formal study of this optimization problem.

Several special cases of the results in section 2 can already be found in the literature. First consider criteria related to the holding costs. Kameda [7] showed that the μc rule is optimal if $\lambda_1 = \cdots = \lambda_N$. Righter [11] concludes that the SIP minimizes the queue length stochastically if $\lambda_1 \leq \cdots \leq \lambda_N$ and $\mu_1 \geq \cdots \geq \mu_N$. This result can also be found in Koole [9].

For the open version of this model (i.e., the model with N customer classes, with class j having Poisson arrival with rate λ_j) the μc rule (i.e., the list policy that orders the customers in decreasing value of $\mu_j c_j$) is well known to be optimal (e.g., [2]). Obviously, the μc rule does not depend on the λ_i.

For criteria related to the idleness of the server results can be found stating that the SIP maximizes the server availability (in expectation, or stochastically) if $\lambda_1 \leq \cdots \leq \lambda_N$, or that the server availability is equal under every policy if $\lambda_1 = \cdots = \lambda_N$. We refer to Courcoubetis et al. [5,4], Kameda [7] and Righter [11].

2. The ordered case

We formulate our model as a Markov decision process. Formally speaking, this amounts to deriving the dynamic programming (dp) equation of the form

$$v_i^{m+1} = \min_{a \in A(i)} \Big\{ k_{ia} + \beta \sum_{i' \in E} p_{iai'} v_{i'}^m \Big\} \quad \text{for all } i \in E. \tag{2.1}$$

In (2.1), E is the state space, $A(i)$ is the set of available actions in state i, k_{ia} are the direct costs, $p_{iai'}$ is the probability of a transition from state i to i' if action a is used in i, and $\beta \le 1$ is the discount factor. If E and $A(i)$ are finite (and under a periodicity condition), it can be shown (see e.g. Tijms [13]) that (for arbitrary v^0) the minimizing policy in v^m converges to the optimal one under the long range discounted cost criterion (or, if $\beta = 1$, under the average cost criterion).

In our problem the state space E consists of all vectors $i = (i_1, \dots, i_n)$ with $i_j = 1$ if customer j is at the queue, and $i_j = 0$ if customer j is functioning. The action spaces $A(i)$ of state i are all the actions j with $i_j = 1$. Furthermore we assume that every $A(i)$ contains action 0, corresponding to idling of the server.

Equation (2.1) is formulated for a discrete time model, ours however is a continuous time one. Thus we need to uniformize our model. Let $\alpha = \sum_{n=1}^N \lambda_n + \mu$ with $\mu = \max_n \mu_n$. Without loss of generality we assume that $\alpha = 1$. By [12], our continuous time model has the same minimizing action as the discrete time model with

$$\begin{aligned} p_{iai+e_n} &= (1 - i_n)\lambda_n \text{ for all } i \in E \text{ and } a \in A(i),\ 1 \le n \le N \\ p_{iai-e_a} &= \mu_a \text{ for all } i \in E,\ a \in A(i),\ i \neq 0 \\ p_{iai} &= 1 - \sum_{n=1}^N (1 - i_n)\lambda_n - \mu_a \text{ for all } i \in E, a \in A(i),\ \mu_0 = 0 \\ p_{iai'} &= 0 \text{ otherwise,} \end{aligned}$$

where $e_l = (0, \dots, 0, 1, 0, \dots, 0)$, with the 0 at the lth position.

We take $v_i^0 = 0$ for all i. Thus the dp equation for our model is

$$\begin{aligned} v_i^{m+1} &= \min_{a \in A(i)} \Big\{ k_i + \mu_a v_{i-e_a}^m + \big(1 - \sum_{n=1}^N (1 - i_n)\lambda_n - \mu_a\big) v_i^m + \sum_{n=1}^N (1 - i_n)\lambda_n v_{i+e_n}^m \Big\} \\ &= k_i + \sum_{n=1}^N (1 - i_n)\lambda_n v_{i+e_n}^m + \sum_{n=1}^N i_n \lambda_n v_i^m + \min_{a \in A(i)} \Big\{ \mu_a v_{i-e_a}^m + (\mu - \mu_a) v_i^m \Big\}. \end{aligned} \tag{2.2}$$

Note that we assume that the direct costs depend only on the state, not on the action. Now we formulate our basic theorem, in which we give conditions on the cost functions for the SIP to be optimal.

Theorem 2.1. *If $\lambda_1 \leq \cdots \leq \lambda_N$ and the cost function k_i satisfies*

$$\mu_{j_1}(k_{i-e_{j_1}} - k_i) \leq \mu_{j_2}(k_{i-e_{j_2}} - k_i) \text{ for } 0 < j_1 < j_2 \text{ and } i_{j_1} = i_{j_2} = 1 \tag{2.3}$$

and

$$k_{i-e_j} \leq k_i \text{ for } i_j = 1, \tag{2.4}$$

then the SIP minimizes (2.2).

A similar result for the open model as discussed in the introduction can be found in Hordijk and Koole [6].

Proof. We prove that

$$\mu_{j_1} v^m_{i-e_{j_1}} + (\mu - \mu_{j_1}) v^m_i \leq \mu_{j_2} v^m_{i-e_{j_2}} + (\mu - \mu_{j_2}) v^m_i,\ 0 < j_1 < j_2 \tag{2.5}$$

and

$$v^m_{i-e_j} \leq v^m_i \tag{2.6}$$

hold for all relevant i and m. From this it follows directly that (2.2) is minimized by the action according to the SIP: by (2.5) we have to serve the customer with the lowest index available, and by (2.6) we should not idle.

The proof is by induction. As $v^0_i = 0$, the basic step is trivial.

Now suppose that (2.6) is proven for $1, \dots, m$, so we need to prove it for $m+1$. First we derive several inequalities; summing them will give the inequality wanted.

First note that $k_{i-e_j} \leq k_i$. For $1 \leq n \leq N$, but $n \neq j$, we have $(i - e_j)_n = i_n$, and it follows by induction that

$$(1 - (i - e_j)_n)\lambda_n v^m_{i-e_j+e_n} + (i - e_j)_n \lambda_n v^m_{i-e_j} \leq (1 - i_n)\lambda_n v^m_{i+e_n} + i_n \lambda_n v^m_i.$$

We also need

$$(1 - (i - e_j)_j)\lambda_j v^m_{i-e_j+e_j} = i_j \lambda_j v^m_i.$$

Suppose that j^* is the optimal action in i and that $j^* \neq j$. Then j^* is also the optimal action in $(i - e_j)$, and follows that

$$\min_{a \in A(i-e_j)} \{\mu_a v^m_{i-e_j-e_a} + (\mu - \mu_a) v^m_{i-e_j}\} = \mu_{j^*} v^m_{i-e_j-e_{j^*}} + (\mu - \mu_{j^*}) v^m_{i-e_j} \leq$$

$$\mu_{j^*} v^m_{i-e_{j^*}} + (\mu - \mu_{j^*}) v^m_i = \min_{a \in A(i)} \{\mu_a v^m_{i-e_a} + (\mu - \mu_a) v^m_i\}.$$

Now consider the case that $j^* = j$. As 0 is an allowable action in $i - e_j$, it follows that

$$\min_{a \in A(i-e_j)} \{\mu_a v^m_{i-e_j-e_a} + (\mu - \mu_a) v^m_{i-e_j}\} \leq \mu_0 v^m_{i-e_j} + (\mu - \mu_0) v^m_{i-e_j} \leq$$

$$\mu_j v^m_{i-e_j} + (\mu - \mu_j) v^m_i = \min_{a \in A(i)} \{ \mu_a v^m_{i-e_a} + (\mu - \mu_a) v^m_i \}.$$

Summing the inequalities obtained so far readily gives $v^{m+1}_{i-e_j} \leq v^{m+1}_i$. Now consider (2.5). We have $\mu_{j_1} k_{i-e_{j_1}} + (\mu - \mu_{j_1}) k_i \leq \mu_{j_2} k_{i-e_{j_2}} + (\mu - \mu_{j_2}) k_i$ if $j_1 < j_2$, by (2.3).

For $1 \leq n \leq N$, but $n \neq j_1, j_2$, we have by induction

$$\mu_{j_1} \lambda_n v^m_{(i-e_{j_1}) \wedge e_n} + (\mu - \mu_{j_1}) \lambda_n v^m_{i \wedge e_n} \leq$$

$$\mu_{j_2} \lambda_n v^m_{(i-e_{j_2}) \wedge e_n} + (\mu - \mu_{j_2}) \lambda_n v^m_{i \wedge e_n},$$

where $\wedge$ denotes the componentwise maximum.

If we consider the terms concerning customers j_1 and j_2, we see that

$$\mu_{j_1} \lambda_{j_1} v^m_i + \mu_{j_1} \lambda_{j_2} v^m_{i-e_{j_1}} + (\mu - \mu_{j_1})(\lambda_{j_1} + \lambda_{j_2}) v^m_i =$$

$$\mu \lambda_{j_1} v^m_i + \mu_{j_1} \lambda_{j_2} v^m_{i-e_{j_1}} + (\mu - \mu_{j_1}) \lambda_{j_2} v^m_i \leq$$

$$\mu \lambda_{j_1} v^m_i + \mu_{j_2} \lambda_{j_2} v^m_{i-e_{j_2}} + (\mu - \mu_{j_2}) \lambda_{j_2} v^m_i =$$

$$\mu \lambda_{j_1} v^m_i + \mu \lambda_{j_2} v^m_i + \lambda_{j_2} \mu_{j_2} (v^m_{i-e_{j_2}} - v^m_i) \leq$$

$$\mu \lambda_{j_1} v^m_i + \mu \lambda_{j_2} v^m_i + \lambda_{j_1} \mu_{j_2} (v^m_{i-e_{j_2}} - v^m_i) =$$

$$\mu_{j_2} \lambda_{j_1} v^m_{i-e_{j_2}} + \mu_{j_2} \lambda_{j_2} v^m_i + (\mu - \mu_{j_2})(\lambda_{j_1} + \lambda_{j_2}) v^m_i,$$

where the first inequality follows from (2.5), the second from the fact that $\lambda_{j_1} \leq \lambda_{j_2}$ and (2.6).

The final step of the proof concerns the departures of the customers in the waiting queue. Suppose that j^* is the optimal action in i and that $j^* \neq j_1$, then $j^* < j_1$ and j^* is also the optimal action in $(i - e_{j_1})$ and $(i - e_{j_2})$. It follows that

$$\mu_{j_1} \min_{a \in A(i-e_{j_1})} \{ \mu_a v^m_{i-e_{j_1}-e_a} + (\mu - \mu_a) v^m_{i-e_{j_1}} \} + (\mu - \mu_{j_1}) \min_{a \in A(i)} \{ \mu_a v^m_{i-e_a} + (\mu - \mu_a) v^m_i \} =$$

$$\mu_{j_1} \{ \mu_{j^*} v^m_{i-e_{j_1}-e_{j^*}} + (\mu - \mu_{j^*}) v^m_{i-e_{j_1}} \} + (\mu - \mu_{j_1}) \{ \mu_{j^*} v^m_{i-e_{j^*}} + (\mu - \mu_{j^*}) v^m_i \} \leq$$

$$\mu_{j_2} \{ \mu_{j^*} v^m_{i-e_{j_2}-e_{j^*}} + (\mu - \mu_{j^*}) v^m_{i-e_{j_2}} \} + (\mu - \mu_{j_2}) \{ \mu_{j^*} v^m_{i-e_{j^*}} + (\mu - \mu_{j^*}) v^m_i \} =$$

$$\mu_{j_2} \min_{a \in A(i-e_{j_2})} \{ \mu_a v^m_{i-e_{j_2}-e_a} + (\mu - \mu_a) v^m_{i-e_{j_2}} \} + (\mu - \mu_{j_2}) \min_{a \in A(i)} \{ \mu_a v^m_{i-e_a} + (\mu - \mu_a) v^m_i \},$$

the inequality by induction. Suppose now that $j^* = j_1$. Because $(i - e_{j_1})_{j_2} > 0$, j_2 is an acceptable action in $(i - e_{j_1})$ and i. Because $(i - e_{j_2})_{j_1} > 0$, j_1 is the optimal action in $(i - e_{j_2})$. Then it follows that

$$\mu_{j_1} \min_{a \in A(i-e_{j_1})} \{ \mu_a v^m_{i-e_{j_1}-e_a} + (\mu - \mu_a) v^m_{i-e_{j_1}} \} + (\mu - \mu_{j_1}) \min_{a \in A(i)} \{ \mu_a v^m_{i-e_a} + (\mu - \mu_a) v^m_i \} \leq$$

$$\mu_{j_1} \{ \mu_{j_2} v^m_{i-e_{j_1}-e_{j_2}} + (\mu - \mu_{j_2}) v^m_{i-e_{j_1}} \} + (\mu - \mu_{j_1}) \{ \mu_{j_2} v^m_{i-e_{j_2}} + (\mu - \mu_{j_2}) v^m_i \} =$$

$$\mu_{j_2} \min_{a \in A(i-e_{j_2})} \{ \mu_a v^m_{i-e_{j_2}-e_a} + (\mu - \mu_a) v^m_{i-e_{j_2}} \} + (\mu - \mu_{j_2}) \min_{a \in A(i)} \{ \mu_a v^m_{i-e_a} + (\mu - \mu_a) v^m_i \}.$$

Summing all terms gives (2.5) for $m + 1$. □

What remains to be done is to study the cost function satisfying (2.3) and (2.4). The first type of cost function we study is $k_i = \sum_n i_n c_n$, i.e., if customer n is waiting to be repaired, it incurs a holding costs c_n. Inserting this form of function in (2.3) and (2.4) readily gives $\mu_{j_1} c_{j_1} \geq \mu_{j_2} c_{j_2}$ and $c_{j_1} \geq 0$. This leads to the following.

Corollary 2.2. *If $\lambda_1 \leq \cdots \leq \lambda_N$ and $\mu_1 c_1 \geq \cdots \geq \mu_N c_N$, then the SIP (i.e., the μc rule) minimizes the average holding and the discounted holding costs.*

A special case is $c_j = 1$ for all j; now the direct costs are equal to the queue length. Thus the SIP minimizes the average queue length if $\lambda_1 \leq \cdots \leq \lambda_N$ and $\mu_1 \geq \cdots \geq \mu_N$.

The second type of cost function we consider is related to the utilization of the server. For this we take $k_i = \mathrm{I}\{i \neq (0, \ldots, 0)\}$. Minimizing this cost function corresponds to maximizing the time the server is idle. Insertion in (2.3) and (2.4) shows that this is indeed an allowable cost function.

Corollary 2.3. *If $\lambda_1 \leq \cdots \leq \lambda_N$, then the SIP maximizes the average (or discounted) time that the server is idle.*

The results presented here can be strengthened in various directions. By taking v^0 as cost function and no direct costs, other objectives as the expected costs at some T can be considered. A rigorous proof of this can be found in [9]. For this objective other cost functions become interesting: $\mathrm{I}\{\sum_n i_n > s\}$ is allowable if $\mu_1 \geq \cdots \geq \mu_N$, giving that the SIP minimizes the queue length stochastically at T if $\lambda_1 \leq \cdots \leq \lambda_N$ and $\mu_1 \geq \cdots \geq \mu_N$. Similarly, the probability that the system is empty at T is maximized by the SIP if $\lambda_1 \leq \cdots \leq \lambda_N$ and $\mu_1 \geq \cdots \geq \mu_N$.

It is shown in [6] that for the open model the cost functions must also satisfy (2.3) and (2.4). There a more detailed study of the allowable cost functions is conducted, including the above results.

3. Counterexample

Let us consider again the holding cost case. In the previous section we found that if both $\lambda_1 \leq \cdots \leq \lambda_N$ and $\mu_1 c_1 \geq \cdots \geq \mu_N c_N$, then a list policy, in this case the μc rule, is optimal. In this section we show that in the general case, neither the μc rule nor any other list policy is optimal. This we do by giving a simple counterexample with only three customers.

For our example we choose a model with the following parameters: $\lambda_1 = 2.00$, $\lambda_2 = 1.00$, $\lambda_3 = 0.10$, $\mu_1 = 3.15$, $\mu_2 = 2.00$, $\mu_3 = 1.00$, $c_1 = 1.00$, $c_2 = 1.00$ and $c_3 = 0.05$. For each of the 24 different policies we computed the average holding costs, using (2.2). For the six list policies the values are given below. Each list policy is characterized by its list, thus policy $\{a, b, c\}$ indicates the policy which gives highest priority to customer a, and lowest priority to c, and its value is denoted by $v(a, b, c)$. The values are as follows: $v(1, 2, 3) = 0.8803$, $v(1, 3, 2) = 0.9338$, $v(2, 1, 3) = 0.8806$, $v(2, 3, 1) = 0.9285$, $v(3, 1, 2) = 0.9569$, and $v(3, 2, 1) = 0.9559$. Thus $(1, 2, 3)$ is the best list policy. However, let us consider the policy that gives lowest priority to the third customer, that serves customer 1 in state $(1, 1, 0)$, but serves customer 2 in state $(1, 1, 1)$. Computations show that this policy is optimal, with value 0.8800. This shows that there need not be an optimal list policy.

We could leave it at that, but let us try to gain some more insight in the model by giving a heuristic explanation for this phenomenon. Customer three plays a role of little importance. It fails seldomly (as $\lambda_3 = 0.10$), en if it has failed, it has the lowest repair priority (as $c_3 = 0.05$). The parameters are chosen such that if only the customers 1 and 2 are available for repair, then customer 1 gets served first. However, if customer 3 is also at the queue, the time it takes to repair customers 1 and 2 plays a more important role, as this determines the instant at which the repair of customer 3 begins. To start repair early on customer 3, service should start with customer 2 (cf. corollary 2.3, as $\lambda_2 < \lambda_1$). The parameters for customer 3 are chosen such that the availability of customers changes the order in which customers 1 and 2 should be served.

4. Asymptotically optimal policies

In section 2 we proved that in some cases the optimal policy for our model is a simple list policy. However, in the previous section we saw that this does not extend to the general case, giving us no hope to find simple optimal policies for each choice of parameters. Therefore we consider in this section the concept of policies which are asymptotically optimal. We restrict ourselves to the holding costs case. In section 4.1 the method is explained. In section 4.2 high traffic is considered, in section 4.3 low traffic.

To avoid trivialities, we assume in this section that $c_j > 0$ for all j. Note that the proof of equation (2.6) did not rely on $\lambda_1 \leq \cdots \leq \lambda_N$. For the holding case this means that action 0 is never optimal. Thus we can as well assume that $0 \notin A(i)$ if $i \neq (0, \ldots, 0)$.

4.1. Description of the method

Our technique for deriving asymptotically optimal policies has first been used (to our knowledge) by Katehakis and Levine [8]. There a routing model under light traffic is considered. The method is strongly related to the power series algorithm, which is a method to derive the stationary distribution of various Markov chains (see e.g. Blanc [1] or Koole [10]).

Starting point of the analysis is the optimality equation, given by

$$v_i = \min_{a \in A(i)} \Big\{ k_{ia} - g + \sum_{i' \in E} p_{iai'} v_{i'} \Big\}, \ i \in E. \tag{4.1}$$

This equation can be used as an alternative to (2.1) to find optimal policies. If E and $A(i)$ are finite there exists a solution with g the average costs, and the minimizing actions give the optimal policy.

Now we replace some of the p_{iaj} by ρp_{iaj} (in our model, we replace for example all λ_i by $\rho\lambda_i$), and we try to derive the optimal policy for small values of ρ. This amounts to minimizing v_i lexicographically. In some cases (as in the models considered here), this asymptotically optimal policy can be derived explicitly.

Let us now consider our model, with holding costs. Equation (4.1) becomes:

$$\begin{aligned} g + v_i = & \sum_{n=1}^{N} i_n c_n + \sum_{n=1}^{N} (1 - i_n)\lambda_n v_{i+e_n} + \Big(1 - \sum_{n=1}^{N} (1 - i_n)\lambda_n - \mu\Big) v_i + \\ & \min_{a \in A(i)} \Big\{ \mu_a v_{i-e_a} + (\mu - \mu_a) v_i \Big\}. \end{aligned} \tag{4.2}$$

This equation has a unique solution if we choose a state i for which we assume $v_i = 0$.

First we consider (4.2) with μ_j replaced by $\rho\mu_j$. After that we consider the case that λ_j is replaced by $\rho\lambda_j$. In both cases we write $g = \sum_{l=0}^{\infty} g^{(l)} \rho^l$ and $v_i = \sum_{l=0}^{\infty} v_i^{(l)} \rho^l$ for all i. Similarly as in [10] it can be shown that in the finite state and action case these power series are well defined and converge for ρ small enough.

4.2. High traffic

We now consider the high traffic case, thus we replace each μ_j by $\rho\mu_j$. Then the following holds.

Theorem 4.1. *If $\mu_1 c_1/\lambda_1 \geq \cdots \geq \mu_N c_N/\lambda_N$ and ρ is small enough, then the SIP minimizes the average holding costs.*

Proof. Substituting the power series of v_i and g in (4.2) (with $\rho\mu_j$ instead of μ_j), and equating corresponding powers of ρ gives

$$g^{(0)} + v_i^{(0)} = \sum_{n=1}^{N} i_n c_n + \sum_{n=1}^{N}(1 - i_n)\lambda_n v_{i+e_n}^{(0)} + \Big(1 - \sum_{n=1}^{N}(1 - i_n)\lambda_n\Big) v_i^{(0)} \tag{4.3}$$

and

$$\begin{aligned} g^{(l)} + v_i^{(l)} &= \sum_{n=1}^{N}(1 - i_n)\lambda_n v_{i+e_n}^{(l)} + \Big(1 - \sum_{n=1}^{N}(1 - i_n)\lambda_n\Big) v_i^{(l)} - \mu v_i^{(l-1)} \\ &\quad + \min_{a \in A(i)} \Big\{\mu_a v_{i-e_a}^{(l-1)} + (\mu - \mu_a) v_i^{(l-1)}\Big\} \text{ for all } l > 0. \end{aligned} \tag{4.4}$$

We see that equations (4.3) and (4.4) constitute a recursive procedure for finding the coefficients of the power series. If we consider (4.3) for $i = (1, \ldots, 1)$, it follows directly that $g^{(0)} = \sum_{k=1}^{N} c_k$. Assuming that $v_{(0,\ldots,0)}^{(0)} = 0$ gives $v_i^{(0)} = \sum_{k=1}^{N} \frac{i_k c_k}{\lambda_k}$, independent of the policy. To find the lexicographically optimal policy we substitute $v_i^{(0)}$ in the minimization term of (4.4) with $l = 1$, and derive the minimizing actions:

$$\min_{a \in A(i)} \Big\{\mu_a (v_{i-e_a}^{(0)} - v_i^{(0)})\Big\} = \min_{a \in A(i)} \{\frac{-\mu_a c_a}{\lambda_a}\}.$$

This policy gives indeed priority to the customer with highest $\mu_j c_j/\lambda_j$ available. □

4.3. Low traffic

In the low traffic case, we replace λ_j by $\rho\lambda_j$. Here we find that the μc rule is optimal.

Theorem 4.2. *If $\mu_1 c_1 \geq \cdots \geq \mu_N c_N$ and ρ is small enough, then the SIP minimizes the average holding costs.*

Proof. Again we substitute the power series of the v_i's and g in (4.2), but now we replace all λ_j by $\rho\lambda_j$. Then we get

$$g^{(0)} + v_i^{(0)} = \sum_{n=1}^{N} i_n c_n + (1 - \mu) v_i^{(0)} + \min_{a \in A(i)} \Big\{\mu_a v_{i-e_a}^{(0)} + (\mu - \mu_a) v_i^{(0)}\Big\} \tag{4.5}$$

and

$$g^{(l)} + v_i^{(l)} = \sum_{n=1}^{N} (1-i_n)\lambda_n v_{i+e_n}^{(l-1)} - \sum_{n=1}^{N} (1-i_n)\lambda_n v_i^{(l-1)} + (1-\mu) v_i^{(l)}$$
$$+ \min_{a\in A(i)} \left\{ \mu_a v_{i-e_a}^{(l)} + (\mu - \mu_a) v_i^{(l)} \right\} \text{ for all } l > 0.$$

Now it suffices just to solve (4.5): the minimizing actions give the asymptotically optimal policy. Equation (4.5) can be seen as the optimality equation of a different model. Because $(0,\dots,0)$ is the single recurrent state, and assuming that $v_{(0,\dots,0)}^{(0)} = 0$, there exists a unique solution to (4.5).

If we consider (4.5) for $i = (0,\dots,0)$, we find that $g^{(0)} = 0$. This simplifies the equation to

$$\min_{a\in A(i)} \left\{ \mu_a (v_{i-e_a}^{(0)} - v_i^{(0)}) \right\} = - \sum_{n=1}^{N} i_n c_n. \tag{4.6}$$

Some tedious calculations show that the solution to (4.6) is given by

$$v_i^{(0)} = \sum_{n=1}^{N} \frac{i_n}{\mu_n} \sum_{l=n}^{N} i_l c_l,$$

and that the minimizing actions are according to the μc rule. □

5. Conclusion

We have studied the scheduling of a single repairman in a finite source. First we have considered list policies. If $\lambda_1 \le \cdots \le \lambda_N$ and $\mu_1 c_1 \ge \cdots \ge \mu_N c_N$ the SIP (the Smallest Index Policy) minimizes the holding costs. Also the server utilization is minimized by the SIP, if $\lambda_1 \le \cdots \le \lambda_N$. Note that this result does not depend on the service rates μ_j. In a similar way we would be able to prove that the server utilization is maximized by the SIP if $\lambda_{\ge} \cdots \ge \lambda_N$.

In section 3 it is shown that the holding costs are not always minimized by a list policy. This motivated us to consider models for which the utilization is either close to 0 or 1. In the first case, i.e., under low traffic, we found that the μc rule is optimal. In the high traffic case the customers with the largest $\mu_j c_j/\lambda_j$ available should be served first. These two policies were found to be among the best list policies in the numerical study conducted in Chakka and Mitrani [3]; our results give a theoretical explanation for their findings.

Acknowledgement. The research of the first author was carried out at CWI, Amsterdam, and supported by the European Grant BRA-QMIPS of CEC DG XIII.

References

[1] J.P.C. Blanc. Performance analysis and optimization with the power-series algorithm. In L. Donatiello and R. Nelson, editors, *Performance Evaluation of Computer and Communication Systems*, pages 53–80. Springer-Verlag, 1993. Lecture Notes in Computer Science 729.

[2] C. Buyukkoc, P. Varaiya, and J. Walrand. The $c\mu$ rule revisited. *Advances in Applied Probability*, 17:237–238, 1985.

[3] R. Chakka and I. Mitrani. Heterogeneous multiprocessor systems with breakdowns: Performance and optimal repair strategies. *Theoretical Computer Science*, 125:91–109, 1994.

[4] C. Courcoubetis and P. Varaiya. Serving process with least thinking time maximizes resource utilization. *IEEE Transactions on Automatic Control*, 29:1005–1008, 1984.

[5] C. Courcoubetis, P. Varaiya, and J. Walrand. Invariance in resource sharing problems. In *Proceedings of the 21th IEEE Conference on Decision and Control*, pages 861–863, 1982.

[6] A. Hordijk and G.M. Koole. On the optimality of LEPT and μc rules for parallel processors and dependent arrival processes. *Advances in Applied Probability*, 25:979–996, 1993.

[7] H. Kameda. A finite-source queue with different customers. *Journal of the ACM*, 29:478–491, 1982.

[8] M.N. Katehakis and A. Levine. A dynamic routing problem—numerical procedures for light traffic conditions. *Applied Mathematics and Computation*, 17:267–276, 1985.

[9] G.M. Koole. *Stochastic Scheduling and Dynamic Programming*. CWI, Amsterdam, 1995. CWI Tract 113.

[10] G.M. Koole. On the use of the power series algorithm for general Markov processes, with an application to a Petri net. *INFORMS Journal on Computing*, 9:51–56, 1997.

[11] R. Righter. Optimal policies for scheduling repairs and allocating heterogeneous servers. *Journal of Applied Probability*, 33:536–547, 1996.

[12] R.F. Serfozo. An equivalence between continuous and discrete time Markov decision processes. *Operations Research*, 27:616–620, 1979.

[13] H.C. Tijms. *Stochastic Modelling and Analysis: A Computational Approach*. Wiley, 1986.