Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 2 Issue 7, July - 2016

Nonpreemptive Scheduling for a Repair Shop with Spares

Abderrahmane Abbou
Dept. of Mechanical & Industrial Eng.
University of Toronto, 5 King's College Road,
Toronto, Ontario, Canada, M5S 3G8
a.abbou@mail.utoronto.ca

Viliam Makis
Dept. of Mechanical & Industrial Eng.
University of Toronto, 5 King's College Road,
Toronto, Ontario, Canada, M5S 3G8
makis@mie.utoronto.ca

***Abstract*—We consider a repair shop serving multiple fleets of failing machines. In order to increase the machine availability, each fleet keeps its own spare machine inventory, which is controlled according to the base-stock policy. The machines fail after operating for an exponentially distributed time with fleet-dependent rate. Upon failure, a machine is immediately sent to the repair shop that can serve one failed machine at a time. Repair times have general probability distributions with fleet-dependent characteristics. The repaired machines, which are as-good-as-new, either join their respective spare machine inventory if the corresponding fleet operates at full capacity, or immediately start operating. The repair shop faces the problem of scheduling the repair of failed machines so as to minimize the sum of inventory holding and machine shortage costs. The system just described is modeled as a single server multiclass finite population queueing system. The optimal scheduling policy, which is restricted to the class of non-idling and nonpreemptive policies, is computed using a semi-Markov decision process. The computational time is considerable for nontrivial systems. A simple heuristic scheduling policy is proposed, which has a near-optimal performance and a short computational time.**

Keywords— Multiclass queueing system; finite population queues; repair shop with spares; server scheduling

I. INTRODUCTION

Capital intensive businesses use various kinds of machines in order to provide various kinds of goods or services. For example, airline companies use fleets of small aircraft for short-distance flights and fleets of large aircraft for long-distance flights. The occasional interruption of operations, due to unexpected machine failures or unscheduled maintenance, have direct consequences on profitability in the form of lost revenue or goodwill. To improve availability, companies maintain spare machines for each fleet, which however comes at the expense of increased inventory holding costs when such spares are ready for use but not put in operation.

Maintenance departments of such companies face the special challenge of assigning repair priorities among the different types of failed machines. They have to deal with various factors such as costs, failure and repair time distributions, on-hand spare inventories, machine shortages, external customers demand, etc. Nevertheless, under certain assumptions, this seemingly complex scheduling problem can be efficiently solved using a simple priority rule.

We model the operations at the maintenance department as a "single server multiclass finite population queueing system". Within this framework, each population corresponds to a fleet of machines as well as the corresponding spares, and the single server corresponds to a repair shop that can serve one failed machine at a time. Thus, broken machines form parallel queues at the service facility, one queue per population, and the server must decide to which queue to switch each time a repair service is completed. This problem is commonly referred to as the "server scheduling problem", which we optimally solve using Markov decision theory. However, it is not easy to obtain and implement the optimal solution for real scheduling problems, and so we propose an effective heuristic approach.

It is important to distinguish between infinite and finite populations when dealing with server scheduling problems. In general, simple (and often static) priority rules are optimal for infinite population models. Particularly, it is optimal to provide service in accordance with the $c\mu$ rule when demand processes are Poisson [1]. Surprisingly, the $c\mu$ rule neither depends on the arrival rates (failure rates in our model) nor on the number of customers (machines in our model) in each queue. For this reason, the $c\mu$ rule has gained much popularity within the queueing theory community. Unfortunately, this rule is not necessarily optimal for finite population models. References [2]-[3] have established certain monotonicity conditions, that we shall discuss later, which ensure the optimality of $c\mu$-like priority rules in finite population settings, but without any consideration of spares.

The models in [2]-[3] are restricted to the exponential repair time distributions. Similar models have been examined in [4]-[6] for general distributions. Reference [7] considered the repair shop problem with spare ma-chine inventories under the exponential repair time assumption. The main contribution of our paper is to extend [7] to the more practical case with generally distributed repair times. This extension is a novel contribution to this important area, which is more realistic than the "memoryless" exponential service time considered e.g. in [7]. Besides the distributional assumption, our paper notably differs from [7] in the calculation of inventory holding costs. Inventory holding costs are charged at a constant rate throughout the infinite planning horizon in [7]. We only charge the holding cost for the machines which are operable but "idling" in the spares inventory. This is a more realistic assumption than charging a holding cost for the operating machines. Moreover, it is usually much more costly to have an idle repaired machine in the spares inventory rather than a failed machine in the repair shop. This is the case, as pointed out by [8], of repairing the navigation systems used by the U.S. Air Force. We also propose a simple heuristic scheduling policy which, under reasonable assumptions, yields near-optimal performance. It is worth mentioning that the special case which considers a single fleet of machines, also referred to as the machine interference problem with spares, has been first studied by [9]. The authors carried out their analysis in the Laplace transform domain using the supplementary variable technique. In contrast, we apply a time-domain approach, which does not require any inverse of the Laplace transform results. Particularly, a Markov renewal theory approach is applied in our paper to derive the expected sojourn times, the expected costs, and the transition probabilities which are required elements for the semi-Markov decision process formulation.

While our research is motivated by maintenance/repair applications, the results of this paper are also relevant for other applications. In fact, our model can be viewed as a "two-stage closed queueing network" which appears in several contexts. For instance, this framework has been used by [10] for modeling performance in computer and communication systems and by [11] for modeling operations in shipbuilding plants. Another important aspect, as discussed in [12], is the interrelationship between the finite population queues with spares and the infinite population queues with finite waiting capacity.

The rest of the paper is organized as follows. In Section II, we formally describe the repair shop scheduling problem. In Section III, we derive the elements of the semi-Markov decision process and present a computational approach based on the value iteration algorithm. Our heuristic scheduling policy is discussed in Section IV. We provide numerical results in Section V and conclude the paper in Section VI.

II. Problem Description

Consider a system with m fleets, indexed by r, and a single repair shop responsible for fleets' maintenance. Define the set $\mathcal{M} = \{1,.,m\}$. Fleet $r \in \mathcal{M}$ can operate up to M_r identical machines at any given time. These machines have exponentially distributed failure times with time-homogeneous rate λ_r. The failed machines are immediately sent to the repair shop where one machine can be served at a time in a nonpreemptive manner. So, the arriving machines must queue for service if the repair shop is busy. Specifically, the machines belonging to fleet r, i.e. type r machines, must wait in queue r, $r \in \mathcal{M}$. Hence, m parallel queues compete for the same repair server. We model the repair time of a type r machine as a random variable T_r with distribution function F_r, density function f_r, and mean $1/\mu_r$. It is assumed that all failure and repair times are mutually independent.

Each fleet r keeps its own spare machine inventory so that a broken machine is immediately replaced by a spare machine, if available in stock. The base-stock policy with parameter $S_r \geq 0$ is used to control such inventory. That is, a repair request is sent to the repair shop as soon as a machine fails in fleet r. This implies that the repair shop cannot idle while repair request(s) are pending. Note that the repair shop can have up to $K_r := M_r + S_r$ repair requests from fleet r. After repair completion, a type r machine joins the spare machine inventory if exactly M_r machines are operating in fleet r. Otherwise, it immediately starts operating. It is assumed that the repaired machines become as-good-as-new, i.e. perfect repair.

We assume the following cost structure which is standard in many inventory-queueing control problems. Fleet r incurs an inventory holding cost at constant rate h_r per spare machine available in stock (up to S_r spares) and a downtime cost at constant rate c_r per machine shortage occurrence (up to M_r shortages). The repair shop seeks a scheduling policy minimizing the expected sum of holding and shortage costs.

The problem considered in this paper naturally arises in many business environments. For example, a typical discrete part manufacturer uses CNCs (computer numeric control machines) and AGVs (automated guided vehicles), among other equipment, for the processing and handling of materials. The throughput of finished products is affected when, e.g., a CNC and/or an AGV breaks down. Hence, the manufacturer keeps safety stock of such critical equipment in order to avoid disruptions. Power generators and fleets of trucks in mining and logistics companies are other examples.

III. Model

In this section, we develop a mathematical model for the repair shop scheduling problem. The long-run average cost is chosen as the optimization criterion, i.e. we aim at minimizing the expected holding plus downtime cost per unit time. The scheduling problem can be formulated as an infinite-horizon semi-Markov decision problem. This framework requires the

specification of (a) the state and policy space, (b) the expected single-stage costs, and (c) the transition probabilities (we refer to [13] for an excellent treatment of Markov decision theory).

A.State Space and Scheduling Policies

Recall that the repair policy is nonpreemptive. This implies that scheduling decisions can only be made at time instances when a failed machine (of any type) completes its repair service or when a failed machine (of any type) finds the repair shop idle upon its failure. These time instances are the decision epochs.

The first step in developing the decision model is to describe the system state at the decision epochs. An intuitive state description is the number of failed machines in each of the m queues. In fact, the repair server would check the length of these queues before deciding which machine type to repair next. Note the irrelevance of the information as to how long each machine has been operating prior to a decision epoch (i.e. the machine age). This is due to the memoryless property of the exponentially distributed machine failure times as well as the perfect repair assumption. Another key implication of the memoryless property is that the stochastic process describing the number of failed machines in each queue at the decision epochs is a Markov renewal process.

Denote the state space by $\mathcal{S}$, where $\mathcal{S} = \{0,., K_1\} \times \cdots \times \{0,., K_m\}$. Hence, $\mathbf{x} \in \mathcal{S}$ is an m-dimensional vector whose rth element, $\mathbf{x}_r$, is the number of failed type r machines. Now, we define the scheduling policies of interest. Let $\mathcal{A} = \{0,1,.,m\}$ be the set of scheduling actions. Here action "0" means do-nothing, action "1" means repair a type 1 machine, action "2" means repair a type 2 machine, and so on. Let $\mathcal{A}(\mathbf{x}) \subset \mathcal{A}$ be the set of permissible scheduling actions in state $\mathbf{x}$. It follows by the non-idleness nature of the repair policy that

$$\mathcal{A}(\mathbf{x}) = \begin{cases} \{0\}, & \mathbf{x} = \mathbf{0}, \\ \bigcup_{r=1}^{m} \{r\} \mathbb{I}\{\mathbf{x}_r > 0\}, & \text{otherwise}, \end{cases} \quad (1)$$

where $\mathbf{0}$ is the m-dimensional zero vector and $\mathbb{I}\{\gamma\}$ is the indicator function, i.e. $\mathbb{I}\{\gamma\} = 1$ if the condition γ is true and $\mathbb{I}\{\gamma\} = 0$ otherwise. For example, if the repair

shop maintains two fleets of machines ($m = 2$) then,

$$\mathcal{A}(\mathbf{x}) = \begin{cases} \{0\}, & \mathbf{x}_1 = 0, \mathbf{x}_2 = 0, \\ \{1\}, & \mathbf{x}_1 > 0, \mathbf{x}_2 = 0, \\ \{2\}, & \mathbf{x}_1 = 0, \mathbf{x}_2 > 0, \\ \{1,2\}, & \mathbf{x}_1 > 0, \mathbf{x}_2 > 0. \end{cases}$$

A stationary scheduling policy $\pi: \mathcal{S} \to \mathcal{A}$ prescribes an action $\pi_{\mathbf{x}} \in \mathcal{A}(\mathbf{x})$ whenever the system is found to be in state $\mathbf{x} \in \mathcal{S}$. The collection of all stationary, non-idling, and nonpreemptive scheduling policies is denoted by Π.

B.Preliminary Queueing Analysis

In this section, we analyze the queue lengths process while some failed machine is undergoing repair. Hence, suppose that $\mathbf{x} \in \mathcal{S} \backslash \{\mathbf{0}\}$ at the current decision epoch and that action $a \neq 0$ is taken, i.e. at least one queue is nonempty and a type a machine is selected for repair at the time origin .

Let $\mathbf{Q}(t) = (\mathbf{Q}_1(t), ., \mathbf{Q}_m(t))$ be the queue lengths vector at time t, where $\mathbf{Q}_r(t)$ denotes the number of failed machines in queue $r \in \mathcal{M}$ at time t. The various queues evolve independently of one another while a particular repair service is in-progress. Therefore, they can be examined in isolation. For each queue, define the stochastic process $\mathbf{Q}_r = \{\mathbf{Q}_r(t) : t \geq 0\}$ on state space $\{0,., K_r\}$ with transition probability function,

$$q^r_{\mathbf{x},z}(t) = \mathbb{P}\{\mathbf{Q}_r(t) = z | \mathbf{Q}(0) = \mathbf{x}\}.$$

The transition function depends on $\mathbf{x}$ only through $\mathbf{x}_r$, and its derivation is deferred to Appendix A.

We examine the random amount of time during which queue r has z machines while the current repair is in-progress. Specifically, we are interested in the first moment of this random duration, $\Psi^r_{\mathbf{x},z}(a)$, which also depends on $\mathbf{x}$ only through $\mathbf{x}_r$. This quantity is essential for later evaluation of the expected single-stage costs.

It is obvious that $\Psi^r_{\mathbf{x},z}(a) = 0$ when $z < \mathbf{x}_r$ because $\mathbf{Q}_r(t)$ is nondecreasing during any repair time. Also, the following equality holds,

$$\sum_{z=\mathbf{x}_1}^{K_1} \Psi^1_{\mathbf{x},z}(a) = \cdots = \sum_{z=\mathbf{x}_m}^{K_m} \Psi^m_{\mathbf{x},z}(a) = \frac{1}{\mu_a}.$$

Let $\mathbb{E}_{\mathbf{x}}$ denote the expectation conditioned on $\mathbf{Q}(0) = \mathbf{x}$.

$$\begin{aligned} \Psi^r_{\mathbf{x},z}(a) &= \mathbb{E}\left\{t^r_{\mathbf{x},z}(a)\right\} = \mathbb{E}\left\{\mathbb{E}\left\{t^r_{\mathbf{x},z}(a) | T_a\right\}\right\} \\ &= \int_0^\infty \mathbb{E}\left\{t^r_{\mathbf{x},z}(a) | T_a = u\right\} f_a(u) du \\ &= \int_0^\infty \mathbb{E}_{\mathbf{x}}\left\{\int_0^u \mathbb{I}\{\mathbf{Q}_r(t) = z\} dt\right\} f_a(u) du \\ &= \int_0^\infty \int_0^u \mathbb{E}_{\mathbf{x}}\{\mathbb{I}\{\mathbf{Q}_r(t) = z\}\} f_a(u) dt du \end{aligned}$$

$$= \int_0^\infty \int_t^\infty \mathbb{E}_{\mathbf{x}} \{\mathbb{I}\{\mathbf{Q}_r(t) = z\}\} f_a(u) du dt$$
$$= \int_0^\infty \mathbb{E}_{\mathbf{x}} \{\mathbb{I}\{\mathbf{Q}_r(t) = z\}\} [1 - F_a(t)] dt$$
$$= \int_0^\infty \mathbb{P}\{\mathbf{Q}_r(t) = z | \mathbf{Q}(0) = \mathbf{x}\} [1 - F_a(t)] dt.$$

We obtain the next useful result,

$$\Psi^r_{\mathbf{x},z}(a) = \int_0^\infty q^r_{\mathbf{x},z}(t)[1 - F_a(t)]dt, \quad (2)$$

for all $\mathbf{x} \in \mathcal{S}\backslash\{\mathbf{0}\}$, $\mathbf{x}_r \le z \le K_r$, $r \in \mathcal{M}$, and $a \in \mathcal{A}(\mathbf{x})$.

C. Expected Single-Stage Costs

If at the current decision epoch the system is found to be in state $\mathbf{x}$ and action $a \in \mathcal{A}(\mathbf{x})$ is taken, then an expected cost $G_{\mathbf{x}}(a)$ would be incurred from the current until the next decision epoch. This is the expected single-stage cost.

For $\mathbf{x} = \mathbf{0}$ (all queues are empty), only inventory holding costs would be charged for an expected duration $1/\sum_{r=1}^m M_r\lambda_r$, i.e. the expected time to a machine failure when the system operates at maximum capacity. We get,

$$G_{\mathbf{x}}(a) = \sum_{r=1}^m S_r h_r / \sum_{r=1}^m M_r \lambda_r, \quad (3)$$

for $\mathbf{x} = \mathbf{0}$ and $a = 0$.

Suppose that $\mathbf{x} \in \mathcal{S}\backslash\{\mathbf{0}\}$ (at least one queue is nonempty) at the current decision epoch and that action $a \neq 0$ is taken. While queue r has z machines, then either an inventory holding cost is continually incurred at rate $(S_r - z)h_r$, or a machine shortage cost is continually incurred at rate $(z - S_r)c_r$. Since queue r would have z machines for an expected duration $\Psi^r_{\mathbf{x},z}(a)$, then we can express $G_{\mathbf{x}}(a)$ as,

$$G_{\mathbf{x}}(a) = \sum_{r=1}^m \sum_{z=\mathbf{x}_r}^{K_r} \left[(S_r - z)^+ h_r + (z - S_r)^+ c_r\right] \Psi^r_{\mathbf{x},z}(a), \quad (4)$$

for all $\mathbf{x} \in \mathcal{S}\backslash\{\mathbf{0}\}$ and $a \in \mathcal{A}(\mathbf{x})$, where $(k)^+ = \max\{k, 0\}$.

D. Transition Probabilities

If at the current decision epoch the system is found to be in state $\mathbf{x}$, then action a would take the system to state $\mathbf{y} \in \mathcal{S}$ at the next decision epoch with probability $P_{\mathbf{x},\mathbf{y}}(a)$.

Assume that all queues are empty at the current decision epoch so that the next decision epoch is triggered by a failed machine arrival. Let $\mathbf{e}_r$ be the m-dimensional vector with value 1 at its rth position and values 0 elsewhere. Then,

$$P_{\mathbf{x},\mathbf{y}}(a) = M_r\lambda_r / \sum_{r'=1}^m M_{r'}\lambda_{r'}, \quad (5)$$

for $\mathbf{x} = \mathbf{0}$, $\mathbf{y} = \mathbf{e}_r$, $r \in \mathcal{M}$, and $a = 0$.

Suppose that at least one queue is nonempty at the current decision epoch. If action $a \neq 0$ is selected, then a transition into state $\mathbf{y}$ implies that $(\mathbf{y}_a + 1 - \mathbf{x}_a)$ type a machines have joined queue a and that $(\mathbf{y}_r - \mathbf{x}_r)$ type r machines, for all $r \neq a$, have joined queue r during the current repair time. This is true because the type a machine undergoing repair would leave the repair shop immediately after service completion. Recall that the processes $\{\mathbf{Q}_r,\ r \in \mathcal{M}\}$ are independent during any repair time. We get,

$$\begin{aligned} P_{\mathbf{x},\mathbf{y}}(a) =& \mathbb{P}\{\mathbf{Q}(T_a) = \mathbf{y} | \mathbf{Q}(0) = \mathbf{x}\} \\ =& \mathbb{P}\{\mathbf{Q}_1(T_a) = \mathbf{y}_1 | \mathbf{Q}(0) = \mathbf{x}\} \\ & \times \cdots \times \mathbb{P}\{\mathbf{Q}_m(T_a) = \mathbf{y}_m | \mathbf{Q}(0) = \mathbf{x}\}, \end{aligned}$$

where for each $r \in \mathcal{M}$,

$$\begin{aligned} & \mathbb{P}\{\mathbf{Q}_r(T_a) = \mathbf{y}_r | \mathbf{Q}(0) = \mathbf{x})\} \\ &= \int_0^\infty \mathbb{P}\{\mathbf{Q}_r(t) = \mathbf{y}_r | \mathbf{Q}(0) = \mathbf{x})\} f_a(t) dt \\ &= \begin{cases} \int_0^\infty q^r_{\mathbf{x},\mathbf{y}_r+1}(t) f_a(t) dt, & r = a, \\ \int_0^\infty q^r_{\mathbf{x},\mathbf{y}_r}(t) f_a(t) dt, & r \neq a. \end{cases} \end{aligned}$$

Consequently, the transition probability is as follows,

$$\begin{aligned} P_{\mathbf{x},\mathbf{y}}(a) = & \int_0^\infty q^a_{\mathbf{x},\mathbf{y}_a+1}(t) f_a(t) dt \\ & \times \prod_{r=1, r\neq a}^m \int_0^\infty q^r_{\mathbf{x},\mathbf{y}_r}(t) f_a(t) dt, \end{aligned} \quad (6)$$

for all $\mathbf{x}, \mathbf{y} \in \mathcal{S}$ such that $\mathbf{x} \neq \mathbf{0}$, $\mathbf{y}_a \ge \mathbf{x}_a - 1$, $\mathbf{y}_r \ge \mathbf{x}_r$ (for all $r \neq a$), and $a \in \mathcal{A}(\mathbf{x})$.

It should be noted that the state transitions which are not in (5) or (6) are not feasible, and accordingly their probabilities are equal to zero.

E. Computing the Optimal Policy

Let g^π be the average cost per time unit that the system would incur in the long-run if the scheduling policy $\pi \in \Pi$ is adopted. That is,

$$\begin{aligned} g^\pi = \lim_{t\to\infty} \frac{1}{t} \mathbb{E}_\pi \Big\{ \int_0^t \sum_{r=1}^m \Big[& (S_r - \mathbf{Q}_r(u))^+ h_r \\ & + (\mathbf{Q}_r(u) - S_r)^+ c_r \Big] du \Big\}, \end{aligned}$$

where $\mathbb{E}_\pi$ denotes the expectation with respect to π. Then, the minimum long-run average cost g^* and the optimal scheduling policy π^* are defined as,

$$g^* = \min_{\pi\in\Pi} g^\pi, \quad (7)$$
$$\pi^* = \operatorname{argmin}_{\pi\in\Pi} g^\pi. \quad (8)$$

One way to solve problem (7), or equivalently problem (8), is using the value iteration algorithm. In fact, each iteration of this algorithm produces bounds

on the minimum average cost g. The algorithm terminates when the relative distance between these bounds is less than a prespecified stopping parameter ϵ. The solutions obtained using this algorithm are called ϵ-optimal solutions. However, the uniformization technique should be applied in order to transform our original continuous-time scheduling problem into a discrete-time counterpart (see [13] for more details).

For all $\mathbf{x} \in \mathcal{S}$, let $v_{\mathbf{x}}^{(n)}$ be the value function associated with state $\mathbf{x}$ at the nth iteration of the algorithm. This quantity can be thought of as the minimum total expected cost that the system would incur if it starts in state $\mathbf{x}$ and n decision epochs are left. It is worth mentioning that these decision epochs could be fictitious in the sense that the state transitions could also be fictitious in the transformed discrete-time problem. The value iteration algorithm for the discrete-time problem is as follows [13]:

Step 0 Obtain $G_{\mathbf{x}}(a)$ using equations (3)-(4) and $P_{\mathbf{x},\mathbf{y}}(a)$ using equations (5)-(6). Set $\tau(a) := 1/\sum_{r=1}^{m} M_r\lambda_r$ if $a = 0$ and $\tau(a) := 1/\mu_a$ otherwise. Set $\tau := \min_{a\in\mathcal{A}} \tau(a)$. Set $v_{\mathbf{x}}^{(0)} := 0$ for all $\mathbf{x} \in \mathcal{S}$. Select a stopping parameter $\epsilon \in [0, 1]$. Let $n := 1$.

Step 1 For each $\mathbf{x} \in \mathcal{S}\backslash\{\mathbf{0}\}$, compute

$$v_{\mathbf{x}}^{(n)} = \min_{a\in\mathcal{A}(\mathbf{x})} \left[\frac{G_{\mathbf{x}}(a)}{\tau(a)} + \frac{\tau}{\tau(a)} \sum_{\mathbf{y}\in\mathcal{S}} P_{\mathbf{x},\mathbf{y}}(a) v_{\mathbf{y}}^{(n-1)} + \left(1 - \frac{\tau}{\tau(a)}\right) v_{\mathbf{x}}^{(n-1)} \right]. \quad (9)$$

Step 2 Calculate the lower and upper bound on g from

$$LB^{(n)} = \min_{\mathbf{x}\in\mathcal{S}} \{v_{\mathbf{x}}^{(n)} - v_{\mathbf{x}}^{(n-1)}\},$$
$$UB^{(n)} = \max_{\mathbf{x}\in\mathcal{S}} \{v_{\mathbf{x}}^{(n)} - v_{\mathbf{x}}^{(n-1)}\}.$$

Step 3 If $\frac{UB^{(n)}-LB^{(n)}}{LB^{(n)}} \leq \epsilon$, then stop. Otherwise, let $n := n + 1$ and go to Step 1.

This algorithm yields the ϵ-optimal average cost $g^* \in [LB^{(n^*)}, UB^{(n^*)}]$. The ϵ-optimal scheduling policy is obtained by setting $\pi_{\mathbf{x}}^*$, for each $\mathbf{x} \in \mathcal{S}$, to the action a minimizing the right side of (9). Note that both g and are also ϵ-optimal for the original continuous-time scheduling problem.

IV. Heuristic Scheduling

The semi-Markov decision approach discussed in the previous section may not be appealing for a couple of reasons. First, implementing the value iteration algorithm requires the knowledge of the exact repair time distributions plus the evaluation of the integrals in (2) and (6). Second, this approach poses computational problem for practical values of m, i.e. the number of fleets, due to the curse of dimensionality. Last, the obtained optimal scheduling policy does not necessarily possess a simple structure, as illustrated by the next example, and hence, it may be difficult to implement.

Example 1 Consider a system with two fleets of machines ($m = 2$). The various parameters are given in the following table, where it is assumed that the re-pair time for both machine types is Erlang with three stages. The policy generated by the value iteration algorithm with $\epsilon := 0.001$ is shown in Figure 1. Indeed, Figure 1 indicates that the obtained policy is not simple in the sense that it is not monotone in the length of queue 2 (i.e. $\mathbf{x}_2$). For instance, when $\mathbf{x}_1 = 8$, the optimal policy selects a type 2 machine for repair if $\mathbf{x}_2 = 5$ and a type 1 machine for repair if $\mathbf{x}_2 = 6$, which is a counterintuitive result.

	M_r	S_r	λ_r	μ_r	h_r	c_r
$r = 1$	6	3	0.2	2.325	0.50	1.22
$r = 2$	9	3	0.3	4.166	0.20	1.00

Figure 1: The optimal scheduling policy for Example 1. The symbol ($\star$) corresponds to the repair of a type 1 machine, and ($\circ$) corresponds to the repair of a type 2 machine.

$\mathbf{x}_2$ \ $\mathbf{x}_1$	0	1	2	3	4	5	6	7	8	9
0		$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
1	$\circ$	$\circ$	$\circ$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
2	$\circ$	$\circ$	$\circ$	$\circ$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
3	$\circ$	$\circ$	$\circ$	$\circ$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$
4	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\star$
5	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\star$
6	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\star$	$\star$
7	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\star$	$\star$
8	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\star$	$\star$
9	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\star$	$\star$
10	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\star$	$\star$	$\star$
11	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\star$	$\star$	$\star$
12	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\star$	$\star$	$\star$

Therefore, a heuristic approach to our scheduling problem is more appropriate. In particular, we seek a policy that can be quickly computed, has managerial appeal, and provides near-optimal performance. As previously mentioned, [3] established conditions under which static priority rules are optimal for the special case: $S_r := 0$ and exponentially distributed T_r for all $r \in \mathcal{M}$. For instance, if the system parameters satisfy the following monotonicity conditions:

C1 $\mu_1 \leq \cdots \leq \mu_m$,

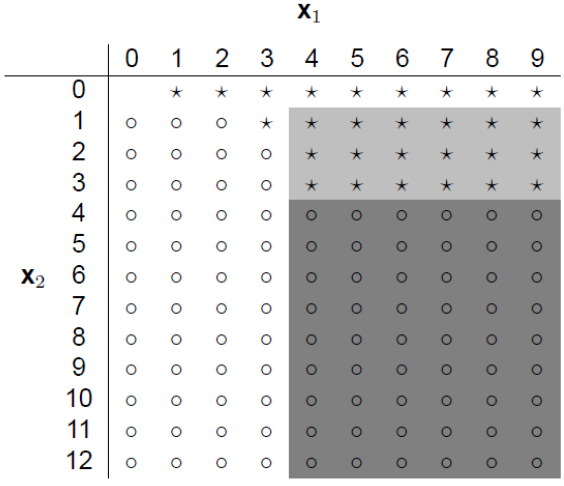
C2 $\lambda_1 \leq \cdots \leq \lambda_m$,

C3 $\frac{c_1\mu_1}{\lambda_1} \leq \cdots \leq \frac{c_m\mu_m}{\lambda_m}$,

then it is optimal to repair the machine type with the highest index $c\mu/\lambda$ among the machine types awaiting re-pair. Call this strategy the $c\mu/\lambda$ rule. Like the $c\mu$ rule, the $c\mu/\lambda$ rule does not depend on the number of failed machines in each queue. Unlike the $c\mu$ rule, the $c\mu/\lambda$ rule depends on the failure rates. This policy has the following intuitive explanation. If all fleets have the same product $c\mu$, then the repair priority is given to the machine type with the smaller failure rate. If all fleets have the same ratio c/λ, then the repair priority is given to the machine type with the shortest processing time. If all fleets have the same ratio μ/λ, then the repair priority is given to the machine type with the highest shortage cost. Observe that Example 1 does not satisfy the monotonicity conditions. Let us show an example that satisfies these conditions.

Example 2 Consider a system similar to that in Example 1 except that λ_2 equals to 0.29 instead of 0.3, so the monotonicity conditions are now satisfied. The policy generated by the value iteration algorithm with ϵ := 0:001 is shown in Figure 2. The obtained policy shows that it is not necessarily optimal to give priority to type 2 machines, although such machines have a higher $c\mu/\lambda$ index. So, the $c\mu/\lambda$ rule is not necessarily optimal for a repair shop scheduling problem with spare machines.

Figure 2: The optimal scheduling policy for Example 2.



Importantly, the $c\mu/\lambda$ rule in Example 2 does not give priority to type 2 machines when the system is short of type 1 machines but not short of type 2 machines, as indicated by the light shaded area in Figure 2. However, it does give priority to type 2 machines when the system is short of both machine types, as indicated by the dark shaded area in Figure 2. For this reason, we suggest a state-dependent $c\mu/\lambda$ rule, so that priority is given to the machine type with the highest $c\mu/\lambda$ index provided the system is indeed short of this machine type. Of course this strategy does not ensure optimality even when the monotonicity conditions hold (one can construct counter-examples).

Now, let us focus on the region of the state space where the system is not short of any machine type (e.g. the shaded area in Figure 1). Our numerical experiments, not shown in this paper, reveal that the optimal solution in this region depends in a complicated way on both the system parameters and the queue lengths. This is not consistent with our quest for an easy-to-implement priority rule. Consider the very special case in which all fleets have the same parameters. Then, it is intuitive to give repair priority to the fleet having the largest number of broken machines at the repair shop. Thus, whenever the system is not facing shortages, we suggest the simple priority rule whereby the server selects the machine type from the longest queue to be re-paired next. We choose the machine type with the lowest inventory holding cost as the tie breaking rule (other criteria are possible). In light of the above discussion, we propose a heuristic scheduling policy, b, defined as follows.

If all queues are empty, i.e. $\mathbf{x} = \mathbf{0}$, then

$$\widehat{\pi}_{\mathbf{x}} := 0.$$

Otherwise, if at least one queue is nonempty and the system is not facing shortages, i.e. $\mathbf{x} \in \mathcal{S}\backslash\{\mathbf{0}\}$ such that $0 \leq \mathbf{x}_r \leq S_r$ for each $r \in \mathcal{M}$, then

$$\widehat{\pi}_{\mathbf{x}} := \underset{r\in\mathcal{M}}{\operatorname{argmin}} \left\{ h_r : r = \underset{r'\in\mathcal{M}}{\operatorname{argmax}}\, \mathbf{x}_{r'} \right\}.$$

Otherwise,

$$\widehat{\pi}_{\mathbf{x}} := \underset{r\in\mathcal{M}}{\operatorname{argmax}} \left\{ \frac{c_r\mu_r}{\lambda_r} : \mathbf{x}_r > S_r \right\}.$$

This policy is dynamic as it depends on the queue lengths **x**, but it uses limited data, namely the failure rates, service rates, holding costs, and shortage costs.

V. Numerical Results

The objective of this section is to test the performance of the proposed policy and to gain further insights into the scheduling problem. We consider systems with three fleets of machines (m = 3) such that $\lambda_1 \leq \lambda_2 \leq \lambda_3$ and $\mu_1 \leq \mu_2 \leq \mu_3$, i.e. conditions **C1** and **C2** hold. Presumably, type 1 (type 3) machines are the most (least) reliable, but at the same time have the longest (shortest) repair time due to their complexity (simplicity). In addition, type 1 (type 3) machines are expensive (cheap) pieces of equipment whose availability is most (least) important. Accordingly, the cost structure is such that $h_1 \geq h_2 \geq h_3$ and $c_1 \geq c_2 \geq c_3$. These assumptions regarding the system parameters are fairly reasonable. However, it is necessary to emphasize that a naïve priority rule giving type 1 (type

3) machines the highest (lowest) priority may not be optimal.

The expected long-run average cost, $\hat{g}$, corresponding to $\hat{\pi}$ can be evaluated through the value iteration algorithm, where (9) becomes

$$v_{\mathbf{x}}^{(n)} := \frac{G_{\mathbf{x}}(\hat{\pi}_{\mathbf{x}})}{\tau(\hat{\pi}_{\mathbf{x}})} + \frac{\tau}{\tau(\hat{\pi}_{\mathbf{x}})} \sum_{\mathbf{y}\in\mathcal{S}} P_{\mathbf{x},\mathbf{y}}(\hat{\pi}_{\mathbf{x}}) v_{\mathbf{y}}^{(n-1)} + \left(1 - \frac{\tau}{\tau(\hat{\pi}_{\mathbf{x}})}\right) v_{\mathbf{x}}^{(n-1)}, \quad (10)$$

for all $\mathbf{x}\in\mathcal{S}$ and $n \geq 1$. Denote by $\tilde{\pi}$ the scheduling policy corresponding to the c/λ rule, i.e. $\tilde{\pi}_{\mathbf{x}} := 0$ if $\mathbf{x} = \mathbf{0}$ and $\tilde{\pi}_{\mathbf{x}} := arg\max_{r\in\mathcal{M}}\{c_r\mu_r/\lambda_r : \mathbf{x}_r > 0\}$ otherwise. Let $\tilde{g}$ be the corresponding long-run average cost. Then $\tilde{g}$ can be evaluated in the same way as $\hat{g}$ by replacing $\hat{\pi}_{\mathbf{x}}$ in (10) with $\tilde{\pi}_{\mathbf{x}}$. Recall that the value iteration algorithm produces bounds $LB^{(n^*)}$ and $UB^{(n^*)}$ at the last iteration n^*. For simplicity, we take the long-run average cost for each policy as the midpoint of the interval $[LB^{(n^*)}, UB^{(n^*)}]$.

In the first experiment, we specify the relevant system parameters in such a way that condition **C3** is also satisfied, i.e. the monotonicity conditions hold. Suppose that h_1 = 0.5, h_2 = 0.4, h_3 = 0.3, c_1 = 1.5, c_2 = 1.2, c_3 = 1.0, μ_1 = 2.7, μ_2 = 4.2, and μ_3 = 5.5. Table 1 shows the calculated average cost for each policy for different combinations of M_r, S_r, and λ_r (all repair times are Erlang random variables with eight stages). Runs 1-3 show that $\hat{\pi}$ and $\tilde{\pi}$ have identical performance when there are no spare machine inventories, which is an obvious result. Most importantly, these runs show that the $c\mu/\lambda$ rule yields optimal solutions. This suggests that the optimality results of in [3] might equally be valid for non-exponential repair/service time distributions (though a more rigorous treatment is required for proving this). Runs 4-9 indicate that the optimality gap is small for our proposed policy and large for the $c\mu/\lambda$ policy. Moreover, these runs reveal an interesting pattern associated with policy $\hat{\pi}$, for which the optimality gap is larger for less congested systems, as illustrated e.g. by run 4 vs. run 6. A plausible explanation for that is as follows. When the congestion level is low, the queue lengths process {**Q**(*t*) : *t* _ 0} would spend considerable time in the region of the state space in which the system is not facing shortages. However, our policy $\hat{\pi}$ is only based on a "coarse" priority assignment in this region of the state space. Thus, sub-optimality is understandably more significant in less congested repair shops. This in turn presents an opportunity for developing a "fine" repair priority assignment when the fleets are not short of machines. For this, it is helpful to thoroughly examine the structure of the optimal policy, which we leave for future work.

Now, let the relevant system parameters be such that the monotonicity conditions do not hold, namely, condition **C3** is violated. Actually, there is no reason for this condition to hold in practice. Suppose that M_1 = 5, S_1 = 4, M_2 = 9, S_2 = 3 M_3 = 2, S_3 = 1, λ_1 = 0.25, λ_2 = 0.30, λ_3 = 0.32, μ_1 = 2.7, μ_2 = 4.2, and μ_3 = 5.5. Table 2 shows the obtained results for different combinations of h_r and c_r (all repair times are Erlang random variables with eight stages). Once again, these results provide strong evidence that our proposed policy is near-optimal and that it outperforms the $c\mu/\lambda$ rule. Also, as the monotonicity conditions break, we see that the performance of $\hat{\pi}$ slightly deteriorates, whereas the performance of $\tilde{\pi}$ severely deviates from the optimum.

VI. Conclusion

In this paper, we have examined the server scheduling problem in a repair shop with spares. The system consists of multiple heterogeneous fleets, where each fleet has both operating and spare machines. Both inventory holding and machine downtime costs have been incorporated. The goal is to assign repair priorities to the failed machines so as these costs are minimized. This problem has been formulated as a semi-Markov decision problem, and the optimal scheduling policy has been numerically computed using the value iteration algorithm. Due to long computational times and implementation difficulties of the optimal policy, we have proposed also a simple heuristic policy with the performance close to the optimum under reasonable assumptions about the given holding and downtimes costs, failure rates, and repair times. As previously mentioned, it would be insightful to examine the structure of the optimal policy in future research. An extension to non-exponential machine failure times would also be desirable as it would provide a more applicable scheduling model.

Acknowledgement

The authors gratefully acknowledge financial support provided by the Natural Sciences and Engineering Research Council of Canada under Grant No. RGPIN 121384-11.

Appendix A - Derivation of the Transition Probability Function

In this section, we express the transition function $q^r_{\mathbf{x},z}(t)$ in closed-form (see equations (11) and (13)-(15)). Recall that the transition function was defined for $\mathbf{x} \in \mathcal{S}\backslash\{\mathbf{0}\}$, i.e. at least one queue is nonempty at the current decision epoch. Let the random variable T be the time to the next decision epoch, e.g. $T \equiv T_a$ if action a is taken at $t = 0$. The key remark is that we only need to define the transition function on time interval $[0, T]$ in order to evaluate (2) and (6). In fact, the $\mathbf{Q}_r$'s are all pure birth processes on this interval. Henceforth, it is assumed that $t \in (0, T)$. We consider two cases depending on the spare machine inventory

Table 1: The long-run average cost for each policy (using $\epsilon := 0.01$) for different M_r, S_r, and λ_r

run	(M_1, M_2, M_3), (S_1, S_2, S_3)	$(\lambda_1, \lambda_2, \lambda_3)$	g^*	$\widehat{g}$	% gap	$\widetilde{g}$	% gap
1	(4,7,9), (0,0,0)	(0.20,0.22,0.24)	3.401	3.401	0	3.401	0
2		(0.25,0.30,0.32)	6.688	6.688	0	6.688	0
3		(0.30,0.35,0.38)	9.046	9.046	0	9.046	0
4	(5,9,2), (4,3,1)	(0.20,0.22,0.24)	2.284	2.342	2.51	2.589	13.33
5		(0.25,0.30,0.32)	3.905	3.943	0.95	4.533	16.06
6		(0.30,0.35,0.38)	6.142	6.150	0.14	6.633	8.01
7	(6,7,3), (0,4,2)	(0.20,0.22,0.24)	2.522	2.526	0.17	3.682	46.02
8		(0.25,0.30,0.32)	4.166	4.168	0.05	5.709	37.05
9		(0.30,0.35,0.38)	6.324	6.324	0	7.552	19.41

Table 2: The long-run average cost for each policy (using $\epsilon := 0.01$) for different h_r and c_r

(c_1, c_2, c_3)	(h_1, h_2, h_3)	g^*	$\widehat{g}$	% gap	$\widetilde{g}$	% gap
(1.8,1.2,1.0)	(0.5,0.4,0.3)	4.133	4.153	0.49	5.843	41.38
	(1.2,0.8,0.4)	4.403	4.497	2.13	8.221	86.71
(2.0,1.5,1.0)	(0.5,0.4,0.3)	4.766	4.815	1.01	6.435	35.02
	(1.2,0.8,0.4)	5.055	5.164	2.15	8.845	74.96
(2.5,2.0,1.0)	(0.5,0.4,0.3)	5.659	5.873	3.78	6.150	8.68
	(1.2,0.8,0.4)	6.044	6.209	2.72	7.266	20.21

Case 1 The spare machine inventory is empty at the time origin, i.e. $\mathbf{x}_r \geq S_r$. In this case, the transition function is known explicitely,

$$q^r_{\mathbf{x},z}(t) = \binom{K_r - \mathbf{x}_r}{z - \mathbf{x}_r} (1 - \exp\{-\lambda_r t\})^{z-\mathbf{x}_r} \times \exp\{-(K_r - z)\lambda_r t\}, \quad (11)$$

for $S_r \leq \mathbf{x}_r \leq K_r$, $\mathbf{x}_r \leq z \leq K_r$, $r \in \mathcal{M}$, and $t \in [0, T]$. Equation (13) is simply the binomial distribution with parameters $(K_r - \mathbf{x}_r)$ and $(1 - \exp\{-\lambda_r t\})$, with the former being the number of operating machines in fleet r at the time origin and the latter being the probability that a type r machine fails by time t.

Case 2 The spare machine inventory is nonempty at the time origin, i.e. $\mathbf{x}_r < S_r$. When it is clear, we suppress the dependence on $\mathbf{x}$ and r for ease of notation. Denote by κ the number of type r machines that have failed in $[0, t]$, i.e. $\kappa = z - \mathbf{x}_r$. Assume that $\kappa > 0$ (the trivial case $\kappa = 0$ is covered by (13)). Let $\theta_1 < \cdots < \theta_\kappa$ be the sequence of failure times. Thus, $(\theta_{l+1} - \theta_l)$ represents the time between the lth failure and the $(l+1)$th failure, for $0 \leq l < \kappa$, where $\theta_0 \equiv 0$. The $(\theta_{l+1} - \theta_l)$'s are all exponentially distributed with failure rate $\Lambda_l = \min\{K_r - \mathbf{x}_r - l, M_r\}\lambda_r$. This follows by the memoryless property of failure times as well as the assumption that spare machines, if available in stock, immediately replace the failed machines. Denote by θ the last failure time, i.e. $\theta \equiv \theta_{\kappa-1}$, and let f_θ be its density function. Hence, θ is the convolution of κ exponential random variables. One approach for obtaining the transition function is through conditioning on θ. This yields,

$$q^r_{\mathbf{x},z}(t) = \mathbb{P}\{\mathbf{Q}_r(t) = z | \mathbf{Q}(0) = \mathbf{x}\} = \int_0^t \mathbb{P}\{\mathbf{Q}_r(t) = z | \mathbf{Q}(0) = \mathbf{x}, \theta = u\} f_\theta(u) du.$$

However, $\mathbb{P}\{\mathbf{Q}_r(t) = z | \mathbf{Q}_r(0) = \mathbf{x}_r, \theta = u\}$ is just the probability that no type r machine breaks down during $[u, t]$. The foregoing integral therefore becomes:

$$q^r_{\mathbf{x},z}(t) = \int_0^t \exp\{-\Lambda(t-u)\} f_\theta(u) du, \quad (12)$$

where $\Lambda = \min\{K_r - \mathbf{x}_r - \kappa, M_r\}\lambda_r$. It can be seen that $\min\{K_r - \mathbf{x}_r - \kappa, M_r\}$ is the number of operating machines in fleet r right after the last failure occurrence. So, the problem reduces to the specification of f_θ. This leads us to the following two subcases.

Case 2-a. Fleet r has M_r operating machines during the entire interval $[0, t]$, i.e. $z \leq S_r + 1$. So, the failure rates are all equal: $\Lambda_0 = \cdots = \Lambda_{\kappa-1} = M_r\lambda_r$. This implies that θ is an Erlang random variable with κ stages. Then,

$$f_\theta(u) = M_r\lambda_r \exp\{-M_r\lambda_r u\} \frac{(M_r\lambda_r u)^{\kappa-1}}{(\kappa-1)!}.$$

The evaluation of (12) yields for $0 \leq \mathbf{x}_r < S_r$, $r \in \mathcal{M}$, and $t \in [0, T]$:

$$f_\theta(u) = M_r\lambda_r \exp\{-M_r\lambda_r u\} \frac{(M_r\lambda_r u)^{\kappa-1}}{(\kappa-1)!}.$$

The evaluation of (12) yields for $0 \leq \mathbf{x}_r < S_r$, $r \in \mathcal{M}$, and $t \in [0, T]$:

$$q^r_{\mathbf{x},z}(t) = (M_r\lambda_r t)^{z-\mathbf{x}_r} \frac{\exp\{-M_r\lambda_r t\}}{(z - \mathbf{x}_r)!}, \quad (13)$$

when $\mathbf{x}_r \leq z \leq S_r$, and

$$q^r_{\mathbf{x},z}(t) = M_r^{z-\mathbf{x}_r} \Big[\exp\{-(M_r - 1)\lambda_r t\} - \sum_{l=0}^{S_r - \mathbf{x}_r} \exp\{-M_r\lambda_r t\} \frac{(\lambda_r t)^l}{l!} \Big], \quad (14)$$

when $z = S_r + 1$.

Case 2-b. Fleet r does not have M_r operating machines during the entire interval $[0,t]$, i.e. $z > S_r + 1$. This means that fleet r has enough spares to operate at full capacity up to the $(S_r + 1 - \mathbf{x}_r)$th failure, but not enough for the last $(z - S_r - 1)$ failures. Thus, $\Lambda_0 = \cdots = \Lambda_{S-\mathbf{x}_r}$ and $\Lambda_{S+1-\mathbf{x}_r} < \cdots < \Lambda_{\kappa-1}$. This implies that θ is the convolution of an Erlang random variable with $S_r + 1 - \mathbf{x}_r$ stages and a hypoexponential random variable with $z - S_r - 1$ stages. Let f_{θ_1} and f_{θ_2} be the density functions of these two random variables, respectively. Then,

$$f_\theta(u) = \int_0^u f_{\theta_1}(v) f_{\theta_2}(u-v)dv,$$

where

$$f_{\theta_1}(v) = M_r\lambda_r \exp\{-M_r\lambda_r v\}\frac{(M_r\lambda_r v)^{S_r-\mathbf{x}_r}}{(S_r-\mathbf{x}_r)!},$$

$$f_{\theta_2}(v) = \sum_{l=S_r+1-\mathbf{x}_r}^{\kappa-1}\left[\prod_{l'=S+1-\mathbf{x}_r, l'\neq l}^{\kappa-1}\frac{\Lambda_{l'}}{\Lambda_{l'}-\Lambda_l}\right]\Lambda_l \exp\{-\Lambda_l v\}.$$

In this case, the evaluation of (12) yields (after some algebra):

$$q^r_{\mathbf{x},z}(t) = \binom{M_r-1}{z-S_r-1}\sum_{l=1}^{z-\mathbf{x}_r}(-1)^{z-S_r-1-l}\binom{z-S_r-1}{l-1} \times\left[\beta\left(\frac{M_r}{z-S_r}\right)^{S_r+1-\mathbf{x}_r} - \beta_l\left(\frac{M_r}{l}\right)^{S_r+1-\mathbf{x}_r}\right], \quad (15)$$

for $0 \le \mathbf{x}_r < S_r$, $S_r+1 < z \le K_1$, $r \in \mathcal{M}$, and $t \in [0,T]$, where β and β_l are given by

$$\beta = \exp\{-(M_r + S_r - z)\lambda_r t\} - \sum_{w=0}^{S_r-\mathbf{x}_r}\exp\{-M_r\lambda_r t\}\frac{((z-S_r)\lambda_r t)^w}{w!},$$

$$\beta_l = \exp\{-(M_r-l)\lambda_r t\} - \sum_{w=0}^{S_r-\mathbf{x}_r}\exp\{-M_r\lambda_r t\}\frac{(l\lambda_r t)^w}{w!}.$$

It should be noted that state transitions which are not given by either (11) and (13)-(15) are not feasible, and accordingly their probabilities are equal to zero.

REFERENCES

[1] J. A. van Mieghem, "Dynamic scheduling with con-vex delay costs: the generalized *cμ* rule," Ann. Appl. Probab., vol. 5, no. 3, pp. 808-833, August 1995.

[2] S. M. R. Iravani and B. Kolfal, "When does the *cμ* rule apply to finite population queueing systems?," Oper. Res. Lett., vol. 33, no. 3, pp. 301-304, May 2005.

[3] S. M. R. Iravani, V. Krishnamurthy, and G. H. Chao, "Optimal server scheduling in nonpreemptive finite population queueing systems," Queueing Syst., vol. 55, no. 2, pp. 95-105, Feb. 2007.

[4] M. J. Chandra and R. G. Sargent, "A numerical method to obtain the equilibrium results for the mul-tiple finite source priority queueing model," Manage-ment Sci., vol. 29, no. 11, pp. 1298-1308, Nov. 1983.

[5] P. Tosirisuk and J. Chandra, "An iterative algorithm for a multiple finite-source queueing model with dy-namic priority scheduling," J. Opl. Res. Soc., vol. 46, no. 7, pp. 905-912, Jul. 1995.

[6] P. Sahba, B. Balcioglu, and D. Banjevic, "Analysis of the finite-source multiclass priority queue with an unreliable server and setup time," Nav. Res. Logist., vol. 60, no. 4, pp. 331-342, Jun. 2013.

[7] W. K. Liang, B. Balcioglu, and R. Svaluto, "Schedul-ing policies for a repair shop problem," Ann. Oper. Res., vol. 211, no. 1, pp. 273-288, Dec. 2013.

[8] K. E. Caggiano, J, A. Muckstadt, and J. A. Rappold, "Integrated real-time capacity and inventory alloca-tion for reparable service parts in a two-echelon sup-ply system," Manufacturing Service Oper. Manage-ment, vol. 8, no. 3, pp. 292-319, Jul. 2006.

[9] U. C. Gupta and T. S.S. Srinivasa Rao, "On the M/G/1 machine interference model with spares," Eur. J. Oper. Res., vol. 89, no. 1, pp. 164-171, Feb. 1996.

[10] J. Sztrik and C. S. Kim, "Markov-modulated finite-source queueing models in evaluation of computer and communication systems," Math. Comp. Mod-elling, vol. 38, no. 7-9, pp. 961-968, Oct. 2003.

[11] F. Dong, J. R. Deglise-Hawkinson, M. P. Van Oyen, and D. J. Singer, "Dynamic control of a closed two-stage queueing network for outfitting process in shipbuilding," Comp. Oper. Res., vol. 72, pp.1-11, Aug 2016.

[12] S. M., Gupta and E. Melachrinoudis, "Comple-mentarity and equivalence in finite source queueing models with spares," Comp. Oper. Res., vol. 21, no. 3, pp. 289-296, Mar. 1994.

[13] H. Tijms, Stochastic Modelling and Analysis: A Computational Approach. Chichester: Wiley, 1986, pp. 190-211.