Manufacturing & Service Operations Management

Workforce Agility in Repair and Maintenance Environments

Seyed M. R. Iravani, Vijayalakshmi Krishnamurthy,

MANUFACTURING & SERVICE OPERATIONS MANAGEMENT
Vol. 9, No. 2, Spring 2007, pp. 168–184
ISSN 1523-4614 | EISSN 1526-5498 | 07 | 0902 | 0168

DOI 10.1287/msom.1060.0132
© 2007 INFORMS

Workforce Agility in Repair and Maintenance Environments

Seyed M. R. Iravani, Vijayalakshmi Krishnamurthy
Department of Industrial Engineering and Management Sciences, Northwestern University,
Evanston, Illinois 60208 {iravani@iems.northwestern.edu, viji.krishnamurthy@philips.com}

In this paper we consider workforce management in repair/maintenance environments in which repairmen are cross-trained to attend more than one type of machine. In this context, we study the machine-repairman problem with heterogeneous machines but with partially cross-trained repairmen. We introduce simple repairman-assignment rules as well as machine-priority rules that are effective in minimizing the machine downtime costs, or balancing the percentage of working machines of different types. We show that static machine priority rules are effective in minimizing systems downtime costs, while a generalized version of the longest queue policy is effective in balancing the percentage of working machines. We also introduce the concept of *hidden symmetry* in repair environments, and show that the well-known chain repairman skill set structure performs very well in repair environments with hidden symmetry. Finally, we provide insights into the design and control issues of repair/maintenance systems with cross-trained repairmen.

Key words: machine-repairman problem; quasi-birth-and-death process; cross-training; preemption; myopic approach
History: Received: January 6, 2003; accepted: June 1, 2006.

1. Introduction

In recent decades, the automation of manufacturing systems, including the introduction of new technologies, has resulted in a wide range of specialized machines, each requiring a different set of repair/maintenance skills. Due to the specialized nature of repair/maintenance operations, companies that sell these machines often provide training programs to the repair/maintenance crew of the companies buying these machines. Consequently, maintenance departments of plants often include a repair crew whose members have different sets of repair skills, whom we refer to as *partially cross-trained repairmen*.

Besides the emerging new technologies that introduce complex machinery, the other reason for having partially cross-trained repairmen involves recruitment and retirement. When an experienced repairman retires, the system often loses a repairman who has a large number of skills. The new hires most likely have a smaller number of skills than the retirees, although they will gradually acquire more skills through various training programs. This interaction of retirement and recruitment results in a partially cross-trained repair crew with some independent and some overlapping sets of skills.

To increase the efficiency of repair systems with partially cross-trained repairmen, managers must answer several important questions such as, which strategy should be followed for assigning repairmen to machines? How should different groups of machines be prioritized in order to minimize the downtime cost of the system or to balance the availabilities of different groups of machines? Which repairmen should be sent for a training program? What is the optimal skill set for the maintenance crew? The existing literature on repair/maintenance problems with heterogeneous machines does not address these questions; instead, it either assumes that there is an unlimited number of repairmen available for each repair operation, or—in the case of a limited number of repairmen—assumes that all repairmen have all the skills required to repair and maintain all types of machines (i.e., that they are all fully cross-trained repairmen). To the best of our knowledge, repair/maintenance environments with partially cross-trained repairmen have not yet been studied. Therefore, to gain insights on the above issues, we focus in this paper on the machine-repairman problem with heterogeneous machines and partially cross-trained repairmen. We develop a model to analyze the performance measures of such systems under a variety of machine-priority and repairman-assignment rules. We provide new insights into the design and control issues of repair/maintenance systems with a partially cross-trained repair crew. We also develop a heuristic algorithm that determines the optimal or near-optimal skill set for each repairman in order to minimize the total average machine downtime and repairman cost.

2. Literature

In the last decade, the utilization of a cross-trained workforce (also referred to as an agile workforce) in manufacturing and service operations has attracted the attention of researchers. For a review of studies on workforce agility in manufacturing and service (e.g., call center) environments, see Hopp and Van Oyen (2004) and Gans et al. (2003). Almost all studies on workforce agility focus on the design and control issues arising in systems with cross-trained workers who are in charge of processing jobs (i.e., production environments). Workforce agility in repair environments, however, has not yet been well analyzed; all that is available are a few studies on machine-repairman problems with fully cross-trained repairmen.

The machine-repairman or machine-interference problem is one of the earliest and most well-known manufacturing applications of operations research in general, and of queueing theory in particular. A large body of literature has been developed in the past three decades that studies a variety of machine-repairman problems. Two reviews of these studies can be found in Stecke and Aronson (1985) and in Stecke (1992).

Within this large body of literature on machine-repairman problems, however, only a few studies deal with the machine-repairman problem with heterogeneous machines. Chandra and Sargent (1983) consider a machine-repairman problem with heterogeneous machines and a single fully cross-trained repairman under general repair time distributions and Poisson failures. They develop a numerical solution procedure to obtain the mean equilibrium results of the performance measures of a multiple finite-source queueing model with fixed nonpreemptive priority discipline. In another paper, Chandra (1986) considers the same problem with Erlang, exponential, and hyperexponential service time distributions to evaluate the effect of service time variability, as well as the fixed nonpreemptive priority and first-come-first-served service discipline, on the performance measures of the system.

Agnihothri (1989) models N heterogeneous machines with K repairmen to develop the interrelationships between the performance measures of the problem with general repair and failure distributions, but without attempting to evaluate each measure. He considers work-conserving service disciplines and examines the performance measures of individual machines and repairmen. Analysis of similar machine-repairman systems with heterogeneous machines and two service stations (consisting of a single-server station and an infinite-server station) under preemptive priority is detailed in Kameda (1982). Kameda obtains an analytic form of such a system under certain conditions and proves that the analytic form is independent of various service disciplines such as first-come-first-served. Koulamas (1992, 1996) derives a scheduling algorithm for a two-machine and single-server system to minimize machine idle time resulting from the unavailability of the server.

System performance measures of a machine-repairman system with one and two fully cross-trained servers are detailed in Shawky (1997). In an attempt to maximize both the machine efficiency and server utilization of a machine-repairman system with a single server, Bahnasawi et al. (1996) use simulation to perform a sensitivity analysis of these measures. Recently, Iravani and Kolfal (2005) and Iravani et al. (2006) studied machine-repairman problems with a single fully cross-trained repairman in systems with preemptive and nonpreemptive repair operations, respectively. They found conditions under which the optimal policy is a static machine priority. Jordan et al. (2004) illustrate the benefit of the chain skill set, a particular type of skill set in which all the repairmen have the same number of skills and all machine types can be served by equal number of repairmen.

They exhibit the robustness of the chain skill set with respect to major changes in system parameters and control decisions.

To the best of our knowledge, there is no literature beyond these studies that discusses machine-repairman problems with partially cross-trained repairmen. This is, therefore, the focus of this paper. In §3, we first investigate the optimal machine-prioritization and repairman-assignments in such systems, and we show that they do not follow a simple rule. We then introduce simple repairman-assignment and machine-priority rules and show how the matrix decomposition approach can be used to analyze systems under these simple rules. In §4, we perform an extensive numerical study to evaluate the performance of our repairman-assignment rules as well as our machine-priority rules from two different perspectives: (i) Minimizing the expected system downtime cost, and (ii) balancing the percentage of working machines among all machine types. Balancing the percentage of working machines of different types is a good criterion in systems where a large number of broken machines of one type may have a significant effect on systems performance (e.g., it creates a sharp bottleneck in a serial production line). Finally, in §4 we provide insights into when the well-known chain cross-training structure provides the most benefit in a repair/maintenance environment with an agile (cross-trained) workforce. Section 5 concludes our paper.

3. Model Description

Consider a machine-repairman problem with N different groups of machines. Group i consists of N_i identical machines with failure rate λ_i ($i = 1, 2, \ldots, N$). The time to failure for all types of machines is assumed to be exponentially distributed, and therefore all machines will have a constant failure rate. This assumption is consistent with the behavior of machines during their useful life phase in the bathtub curve (Kececioglu 1991). The repair rate for machines of type i is μ_i, and is independent of which repairman is performing the repair operation ($i = 1, 2, \ldots, N$). This assumption is a restrictive assumption in cases in which some repairmen are significantly faster than others in repairing machines of type i. We assume that the repair operation, if needed, can be preempted, so that the repairman can be assigned to another (more important) repair operation. We consider exponential distribution for the repair times. Exponentially distributed repair times represent cases where shorter repair times are more probable than longer repair times. In other words, most of the machine failures are minor failures that can be fixed in a short time, while major failures that require an overhaul repair (with longer repair times) are less frequent.

We assume there are K repairmen $1, 2, \ldots, K$, each with a set of repair skills corresponding to the types of machines she is trained to repair. We define *skill set matrix* M of size $K \times N$, with binary elements $m_{k,i}$, where $m_{k,i} = 1$ implies that the repairman k has the skill to repair machines of type i, and $m_{k,i} = 0$ implies otherwise. Let $\|M\|$ be the total number of skills (i.e., the total number of 1s) in the skill set matrix M, and define set $\mathcal{M}_k$ to represent the set of skills of repairman k. Also define $\mathcal{M}^i$ as the set of repairmen who have the skill to repair machines of type i. Let κ_k be the total number of skills of repairman k where $1 \le \kappa_k \le N$. Therefore, for a system with fully cross-trained repairmen, $m_{k,i} = 1$, $\kappa_k = N$, $\|M\| = KN$; $\mathcal{M}^i = \{1, 2, \ldots, K\}$, $\mathcal{M}_k = \{1, 2, \ldots, N\}$ for all $k = 1, 2, \ldots, K$ and $i = 1, 2, \ldots, N$.

In systems with fully cross-trained repairmen, since every repairman has the same number of skills (i.e., N skills), the performance measures of the system (e.g., average machine downtime) are independent of the order in which repairmen are assigned to repair the broken machines. This is because all the repairmen are functionally identical since they possess the same set of skills. However, when the repairmen are only partially cross-trained, the system's performance is significantly influenced by the strategy used to assign repairmen to broken machines. In the next section, we study the complex behavior of the optimal dynamic repairman-assignments to machines.

3.1. The Optimal Policy

The problem can be formulated as a Markov decision process (MDP) as follows:

- System state consists of vectors $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, where x_i is the number of broken machines of type $i \in \{1, 2, \ldots, N\}$, and $0 \le x_i \le N_i$.
- Decision epochs are (i) failure-completion epochs, and (ii) repair-completion epochs.

- Action space $\mathbf{\Phi}(x)$ at state $\mathbf{x}$ includes repairman-assignment vectors $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_K(\mathbf{x}))$, where $\phi_k(\mathbf{x})$ is the machine type to which repairman k is assigned ($\phi_k(\mathbf{x}) \in \mathcal{M}_k$), and $\phi_k = 0$ indicates that repairman k is being kept idle.

Let

$$\Lambda = \sum_{i=1}^{N} N_i \lambda_i; \qquad \Upsilon = \sum_{i=1}^{N} U_i \mu_i; \qquad U_i = \min\{\|\mathcal{M}^i\|, N_i\},$$

where $\|\mathcal{M}^i\|$ is the total number of repairmen who are trained to repair machine type i. Let $\mathbf{1}^i$ be an N element row vector with 1 on its ith element, and 0 in other places, $\mathbf{1}^i = (0, 0, \ldots, 0, 1, 0, \ldots, 0)$; then the optimality equation of the MDP model is as follows (see Lippman 1975):

$$\begin{aligned}
&\frac{g}{\Lambda+\Upsilon} + V(\mathbf{x}) \\
&= \frac{1}{\Lambda+\Upsilon}\Bigg(C(\mathbf{x}) + \sum_{i=1}^{N}(N_i - x_i)\lambda_i V(\mathbf{x}+\mathbf{1}^i) + \sum_{i=1}^{N} x_i \lambda_i V(\mathbf{x}) \\
&\qquad + \min_{\forall \phi(\mathbf{x}) \in \Phi(\mathbf{x})} \Bigg\{ \sum_{i=1}^{N} \|\phi_i(\mathbf{x})\| \mu_i V(\mathbf{x}-\mathbf{1}^i) \\
&\qquad\qquad + \sum_{i=1}^{N} (U_i - \|\phi_i(\mathbf{x})\|)\mu_i V(\mathbf{x}) \Bigg\}\Bigg), \qquad (1)
\end{aligned}$$

where $\|\phi_i(\mathbf{x})\|$ represents the total number of repairmen who are assigned to machines type i under action $\phi(\mathbf{x})$. Note that $C(\mathbf{x})$ is the cost during the transition when action $\phi(\mathbf{x})$ is taken at state $\mathbf{x}$. In this paper, we consider two cost functions that are separable among machines and are linear in the number of broken machines. One example of such functions is $C(\mathbf{x}) = \sum_{i=1}^{N} x_i$, which results in the minimization of the total average number of broken machines in the system.

The optimal solution to (1), which we refer to as the global optimal policy, can be viewed as a combination of two decision rules: (i) machine-priority rules, and (ii) repairman-assignment rules. When a repairman completes a repair, a machine-priority rule specifies the repairman's next repair task, i.e., the next machine type that he should start repairing. On the other hand, when a machine fails, a repairman-assignment rule specifies which repairman among those who have the required skill should be assigned to that machine. One can consider machine-priority rules as rules that give priority to broken machines waiting for repairmen, while repairman-assignment rules give priority to repairmen for performing the repair operation on those machines. Note that when all repairmen are fully cross-trained, repairman-assignment rules would not matter (i.e., all repairmen would have the same priority); however, a machine-priority rule is still essential even in systems with fully cross-trained repairmen.

The optimal solution to (1) has a complex structure and does not follow a simple machine-priority rule or repairman-assignment rule. For example, suppose that the objective is to minimize the total average number of broken machines in the system (i.e., $C(\mathbf{x}) = \sum_{i=1}^{N} x_i$). To study the machine-priority rule under the optimal policy, consider Case A for a system with four machine types, where $(N_1, N_2, N_3, N_4) = (10, 3, 3, 4)$, $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (40, 30, 20, 15)$, $(\mu_1, \mu_2, \mu_3, \mu_4) = (160, 120, 80, 60)$, and the repairman skill set matrix is

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Note that Repairman 1 in matrix M is fully cross-trained and can repair all machine types. Thus, analyzing the assignment of this repairman under the global optimal policy will provide some insights on how that policy prioritizes machine repairs.

Solving (1) for Case A, we find that the optimal policy that minimizes the total average number of broken machines dictates the decisions in Table 1 for states $\mathbf{x} = (3, 3, 1, 1)$, and $\mathbf{x}' = (2, 3, 1, 1)$.

As Table 1 shows, at state $\mathbf{x}$, Repairman 1, who can repair all four machine types, is assigned to a machine of type 1, while broken machines of types 4 (and 2) remain unattended in the system. This may lead to the conjecture that either machines of type 1 have a higher priority than types 4 (and 2), or Repairman 1 is following the longest queue (LQ) policy and thus repairs the machine type with the largest number of broken machines. However, the assignment under the optimal policy for state $\mathbf{x}'$ disproves both of these conjectures. At state $\mathbf{x}'$, Repairman 1 is assigned to a machine of type 4. This confirms that, under the objective of minimizing the average number of broken

Table 1 **Optimal Policy for States x and x' in Case A**

	No. of broken machines				Repairman-assignments to machines			
State	x_1	x_2	x_3	x_4	Repairman 1	Repairman 2	Repairman 3	Repairman 4
x	3	3	1	1	Mach. 1	Mach. 2	Mach. 3	Mach. 1
x'	2	3	1	1	Mach. 4	Mach. 2	Mach. 3	Mach. 1

machines, (i) the optimal policy does not follow a static machine-priority rule that always gives higher priority to machines of type 1 over type 4, and (ii) the optimal policy does not follow the LQ policy. Note that at state $\mathbf{x}'$ machines of type 2 have the longest queue, but Repairman 1 repairs the machine of type 4. We have also studied the repairman-assignment rule under the optimal policy that minimizes the total average number of broken machines in the system; there, too, we did not observe a simple rule. For instance, suppose that in Case A the repairman skill set matrix is M_5, as presented in the Appendix A.[1] We call this scenario Case B. In Case B, there exist 220 states in which exactly two machines of type 2 are broken (i.e., states $\mathbf{x} = (x_1, 2, x_3, x_4)$). Table 2 shows how the optimal policy assigns Repairmen 1, 2, 3, and 4 ($R1$, $R2$, $R3$, and $R4$, respectively) to repair the two broken machines of type 2.

As Table 2 depicts, even when exactly two machines of type 2 are broken, the optimal policy assigns (gives different priorities to) different repairmen to repair those machines. We could not deduce any simple rule for how these two repairmen (and sometimes only one repairman) are chosen. In fact, in experimenting with the optimal policy under different objectives (e.g., minimizing total average downtime cost), we found that the repairman-assignment rule under the optimal policy is very sensitive to the skill set matrix, as well as to the number of broken machines of all types.

The optimal policy, as shown in Cases A and B, has a very complex structure and does not follow simple rules. This observation motivated us to seek an answer to the following question: Is there any simple machine-priority rule or repairman-assignment rule that has close to optimal performance? In the next sections, we first introduce simple machine-priority and repairman-assignment rules, and then we show that implementing policies that consist of these simple rules can result in a performance close to the performance under the optimal policy. We focus on two performance criteria:

- Minimizing the total average machine downtime cost: Considering c_i as the downtime cost per unit time of a machine of type i, for this criterion we have $C(\mathbf{x}) = \sum_{i=1}^{N} c_i x_i$. In practice c_i depends on the effects of downtime of a machine of type i on the production/throughput loss. Note that when $c_i = 1$ for all $i = 1, 2, \ldots, N$, this criterion reduces to minimizing the total average number of broken machines in the system.
- Balancing the percentage of working machines of different types: One way to create a balance among percentage of working machines is to minimize the maximum percentage of broken machines among all machine types. For this case, we have $C(\mathbf{x}) = \max_i\{x_i/N_i\}$. In the rest of the paper, we refer to this criterion as *balancing the percentage of working machines*.

3.2. Machine-Priority Rules

In this section we introduce three different machine-priority rules that are easy to implement in practice, namely (i) the $c\mu$ rule, (ii) the $c\mu/r\lambda$ rule, and (iii) the highest percentage broken first (HPB) rule.

Table 2 **Repairman-Assignment to Type 2 Machines Under the Optimal Policy in Case B**

Number of states (out of 220)	7	8	5	132	13	55
Repairmen assigned to type 2 machine	$R1$ & $R2$	$R1$ & $R4$	$R2$ & $R3$	$R3$ & $R4$	$R2$ & $R4$	$R4$

[1] All appendixes to this paper are available at http://www.informs.org/Pubs/Supplements/MSOM/1526-5498-2007-09-02-0168-ecompanion.pdf.

3.2.1. The $c\mu$ Rule. The $c\mu$ rule has been proven to be optimal in prioritizing tasks in many production and service environments. Cox and Smith (1961) prove the optimality of the $c\mu$ rule in $M/G/1$ queues with linear waiting costs. Van Mieghem (1995) extends the analysis to $G/G/1$ queues with convex costs and shows that the generalized $c\mu$ rule is asymptotically optimal in heavy traffic. Mandelbaum and Stolyar (2004) study parallel open queueing systems with heterogenous cross-trained servers in heavy traffic. They show that, for convex cost functions, the generalized $c\mu$ rule is asymptotically optimal.

We adopt the simple $c\mu$ rule to prioritize repair tasks in our closed queueing system. In our repair environment, the $c\mu$ rule gives the highest priority to the machine type that has the highest value of $c_i\mu_i$. The intuitive reasoning is that this will decrease the downtime cost of the system by working on the machines with higher downtime cost (c_i) and faster repair rate (μ_i). Furthermore, it also releases repairmen faster because they finish more repairs per unit time.

3.2.2. The $c\mu/r\lambda$ Rule. As we mentioned, the intuition behind the $c\mu$ rule is that it tends to reduce the total downtime cost of the system faster by giving priority to machines with higher downtime cost and faster repair rates. However, if the machines with higher downtime cost and faster repair rates also have larger failure rates, then when they are assigned higher priority they will receive their repairs faster, and will therefore fail and come back to the system faster.[2] Hence, assigning high priority to these machines may not accomplish the objective of reducing the total downtime cost of the system. The $c\mu/r\lambda$ rule takes this into account by prioritizing machines according to the following rule: The machines are prioritized based on the ratio $c_i\mu_i/r_i\lambda_i$, where r_i is the total number of repairmen who have the skill to repair machine of type i (i.e., $r_i = \|\mathcal{M}^i\|$). Higher priorities are assigned to machines with higher ratios, while lower priorities are assigned to machines with lower ratios.

[2] Note that the machine-repairman problem is a finite-population queueing system. When a repair is completed, the population of working machines increases, and thus the total failure rate increases.

Note that the $c_i\mu_i/r_i\lambda_i$ rule tends to give priority to machines with (i) higher downtime cost and faster repair rates, which reduces the total downtime cost of the system faster; (ii) smaller failure rates, which reduces the average number of failures per unit time, and therefore the average number of broken machines in the system; and (iii) machines that can be repaired by fewer repairmen, which ensures that those machines will get enough attention.

3.2.3. The HPB Rule. The objective of both $c\mu$ and $c\mu/r\lambda$ machine-priority rules is to minimize the total downtime cost of the system. Here we introduce the HPB rule that tends to balance the percentage of working machines by giving higher priorities to machines with a larger percentage of broken machines.

According to the HPB rule, when a failure occurs or a repair is completed, the broken machines are prioritized based on the percentage broken in each type. Specifically, the machine types are prioritized such that the machine type with the largest percentage broken (i.e., type $j = \arg\max_i\{x_i/N_i\}$) has the highest priority, and the machine type with the least percentage of broken machines (i.e., machine of type $l = \arg\min_i\{x_i/N_i\}$) has the lowest priority. When the number of machines of different types are the same ($N_i = N$ for all i), the HPB rule becomes the LQ policy.

3.3. Repairman-Assignment Rules

As we mentioned in previous sections, when a machine fails a repairman-assignment rule specifies which repairman among those who have the required skill should be assigned to that machine. In §3.3 we introduce four different repairman-assignment rules: (i) the least skilled repairman (LSR) rule, (ii) the least valued repairman (LVR) rule, (iii) the least low priority (LLP) rule, and (iv) the least repair request (LRR) rule. These rules are designed for systems under a static machine-priority rule. Under a static machine-priority rule, a given machine type always has higher priority over another type, regardless of the system state (i.e., the number of broken machines of each type). Without loss of generality, we assume that under a static machine-priority rule, machine types are numbered such that type i is of higher priority than type j if $i < j$. The reason we focus only on static machine-priority rules is that, as we will show in §4, static machine-priority rules perform as well as the dynamic machine-priority rule that the global optimal policy follows.

3.3.1. The LSR Rule. According to the LSR rule, when a machine of type j fails, among all repairmen who have the skill to repair type j machines and are not busy with machines of type j or with higher priority machines, the repairman with the smallest total number of skills (i.e., the least κ_k) is assigned to that task. If there is more than one repairman with the least κ_k value, then the repairman without the skill for repairing type 1 machines (i.e., the machines with the highest priority) is assigned. If all the repairmen with the least κ_k value can also repair type 1 machines, then the repairman who cannot repair type 2 machines (i.e., machines with the second highest priority) is assigned. This process is continued in order of machine-priority until the tie is broken.

The logic behind the LSR rule is to keep the repairmen with more skills available to best utilize the flexibility offered by these highly skilled repairmen. Moreover, the tiebreaking rule attempts to reduce the number of preemptions of lower priority machines by keeping available the repairmen with higher priority skills (i.e., repairmen with skills required to repair higher priority machines).

3.3.2. The LVR Rule. The LSR rule does not consider machine downtime costs when it assigns repairmen to machines. Consider repairman A who has skills to repair machines of types 1 and 2, and repairman B who can repair machines of types 1, 3, and 4. If a machine of type 1 requires repair, the LSR rule assigns repairman A to that machine and keeps repairman B free for future repair tasks. This may not be a good policy if machines have significantly different downtime costs. For example, if the machines downtime costs are $c_1 = c_3 = c_4 = 10$, and $c_2 = 100$, then it makes sense to assign repairman B to repair type 1 machine, and keep repairman A free for potential and more costly failure of type 2 machines. The LVR rule takes this into account.

According to the LVR rule, when a machine of type j fails, among all repairmen who have the skill to repair that machine and are not busy with machine type j or with higher priority machines, repairman R, who has the smallest value of $\sum_{i=1}^{N} m_{k,i} c_i$, is assigned to that task (i.e., $R = \arg\min_{k \in \mathcal{M}^j} \{\sum_{i=1}^{N} m_{k,i} c_i\}$). In case there is more than one repairman with the same least $\sum_{i=1}^{N} m_{k,i} c_i$ value, a tiebreaking rule similar to that in LSR is followed.

The logic behind the LVR rule is to keep the repairmen with more skills or with more valuable skills (or both) available in case machines with higher downtime costs fail.

3.3.3. The LLP Rule. Under a preemption priority scheme, the failure of a higher priority machine always affects the assignment of repairmen to lower priority machines. Therefore, in assigning repairmen to machines, the total number of higher priority skills of a repairman may not by itself create a flexible and efficient repairman assignment. For example, consider a three-machine system where repairman $R1$ has Skills 1 and 2, and repairman $R2$ has Skills 2 and 3. The downtime cost of the machines are $c_1 = 10$, $c_2 = 5$, and $c_3 = 4$. If a machine of type 2 fails, the LVR rule assigns repairman $R2$ to that machine because $\sum_{i=1}^{N} m_{k,i} c_i$ of repairman $R2$ is 9, whereas the $\sum_{i=1}^{N} m_{k,i} c_i$ of repairman $R1$ is 15. Under this assignment, if a machine of type 3 fails, the idle repairman $R1$ cannot be assigned to the type 3 machine because she does not possess the required skill. On the other hand, the repair operation on the machine of type 2 cannot be interrupted since it has a higher priority than type 3. Therefore, the failed type 3 machine will not receive a repair immediately. On the failure of a type 2 machine, however, if the LVR rule is ignored so that repairman $R1$ is assigned to that machine, then on the failure of a machine of type 3, repairman $R2$ will be available to attend that machine. Note that under this assignment, since preemption is allowed, repairman $R1$ can always be immediately called for the repair of a higher priority machine (i.e., higher than 2).

As the above example shows, a flexible repairman-assignment rule should take into account the availability of repairmen for lower priority machines. This is because, under any assignment rule, when preemption is allowed, the higher priority machine will always be repaired before lower priority machines. Thus, to keep repairmen with lower priority skills available for future failures of lower priority machines, when a machine of type i fails, the LLP rule calculates the index θ_k^i for all repairmen $k \in \mathcal{M}^i$:

$$\theta_k^i = \sum_{j>i}^{N} m_{k,j} c_j; \quad k \in \{1, \ldots, K\}.$$

Note that θ_k^i is an indication of the value of repair skills that repairman k possesses for machines of lower priority than i. The LLP rule therefore chooses the repairman with the lowest index θ_k^i. If there is more than one repairman with the lowest index, then the repairman who does not have the skill to repair the next priority machine type (i.e., $i+1$) is chosen for the task.

3.3.4. The LRR Rule. The LSR, LVR, and LLP rules use repairmen's skill sets to determine repairman-assignments, and they try to keep some repairmen available for future failures. They do not, however, take into account the possibility of those future failures. The LRR rule attempts to obtain an estimate for the average number of failures per unit time for each type of machine by assuming unlimited repairmen for each type. Under this assumption, the availability of a type i machine is $\alpha_i = \mu_i/(\lambda_i + \mu_i)$, which implies that there will be, on the average, $N_i\alpha_i$ working machines, which in turn results in an estimate of $\lambda_i N_i \alpha_i$ failures per unit time. Therefore, for each repairman k, the index β_k will be a relative indicator of the weighted average of the number of failures per unit time that require his skills. Those failures are weighted by the relative cost of failures:

$$\beta_k = \sum_{i=1}^{N} m_{k,i} \lambda_i N_i \alpha_i c_i; \quad k \in \{1, 2, \ldots, K\}.$$

When a failure of type j occurs, the LRR rule chooses the repairman with the least value of β_k from among the repairmen who have skill j and are not busy with machines of type j or of higher priority. This keeps those repairmen who can respond to a greater and more expensive range of repair requests free for later use.

Now that we have introduced our repairman-assignment rules, in the next section we develop an analytic model to obtain the performance measures of the machine-repairman problem with a preemptive priority scheme and partially cross-trained repairmen. Our model is general and can be used under any repairman-assignment rule $\mathcal{R} \in \{LSR, LVR, LLP, LRR\}$ and any static machine-priority rules.

3.4. Problem Formulation

Define $x_i(t)$ as the number of failed machines of type i at time t; then the stochastic process $\mathbf{X}(t) = \{x_1(t), x_2(t), \ldots, x_N(t), t \geq 0\}$ will be a continuous time Markov chain (CTMC) with finite state space $\mathcal{X} = \{\mathbf{x} \mid \mathbf{x} = (x_1, x_2, \ldots, x_N);\ 0 \leq x_i \leq N_i,\ \text{for all } i\}$.

Let $\eta_{\mathcal{R}}^{x_i}$ be the number of repairmen busy with the type i machine at state $\mathbf{x}$ based on the rule $\mathcal{R} \in \{LSR, LVR, LLP, LRR\}$. Then the transition rates $q_{\mathbf{x},\mathbf{x}'}$ of the transition rate matrix $\mathbf{Q}$ for the CTMC are as follows:

$$q_{\mathbf{x},\mathbf{x}'} = \begin{cases} \eta_{\mathcal{R}}^{x_i}\mu_i; & \text{if } \mathbf{x}' = \mathbf{x} - \mathbf{1}^i,\ x_i > 0, \\ & i = 1, 2, \ldots, N,\ \text{for all } \mathcal{R}. \\ (N_i - x_i)\lambda_i; & \text{if } \mathbf{x}' = \mathbf{x} + \mathbf{1}^i,\ x_i < N_i, \\ & i = 1, 2, \ldots, N. \\ -\sum_{\mathbf{x}'' \in \mathcal{X},\, \mathbf{x}'' \neq \mathbf{x}} q_{\mathbf{x},\mathbf{x}''}; & \text{if } \mathbf{x} = \mathbf{x}'. \\ 0; \quad \text{otherwise.} \end{cases} \tag{2}$$

Each state in the CTMC communicates with other states through a set of failure or repair events. From this, we can prove that this CTMC is irreducible, and hence its steady state probability distribution $\mathbf{P}$ exists and is unique. To find these steady state probabilities, the direct approach to solve the $\prod_{j=1}^{N}(N_j+1)$ balance equations using the Gaussian elimination method requires $[\prod_{j=1}^{N}(N_j+1)]^3/3 + [\prod_{j=1}^{N}(N_j+1)]^2$ multiplication operations. However, if the states of the CTMC are arranged in a certain way, the matrix decomposition approach can be employed that reduces the number of computations, and presents a simple form solution. Specifically, if the states of the system are arranged in order $\mathcal{X} = \{\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{N_1}\}$, where $\mathbf{X}_j$ is the set of states in which there are j broken machines of type 1 (i.e., $x_1 = j$), and

$$\begin{aligned} \mathbf{X}_j = \{&(j,0,\ldots,0,0), \ldots, (j,0,\ldots,0,N_N), \\ &(j,0,\ldots,1,0), \ldots, (j,0,\ldots,1,N_N), \ldots, \\ &(j,0,\ldots,N_{N-1},0), \ldots, (j,0,\ldots,N_{N-1},N_N), \ldots, \\ &(j,N_2,\ldots,N_{N-1},0), \ldots, (j,N_2,\ldots,N_{N-1},N_N)\}, \end{aligned}$$

then the transition rate matrix **Q** for our CTMC will be as follows:

$$\mathbf{Q}=\begin{bmatrix} A_0 & N_1\lambda_1 I & 0 & 0 & \cdots & 0 & 0 & 0 \\ \mu_1 I & A_1 & (N_1-1)\lambda_1 I & 0 & \cdots & 0 & 0 & 0 \\ 0 & 2\mu_1 I & A_2 & (N_1-2)\lambda_1 I & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & r_1\mu_1 I & A_{r_1} & (N_1-r_1)\lambda_1 I & \cdots & 0 & 0 \\ 0 & 0 & \cdots & r_1\mu_1 I & A_{r_1+1} & (N_1-r_1-1)\lambda_1 I & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & r_1\mu_1 I & A_{N_1-1} & \lambda_1 I \\ 0 & 0 & 0 & 0 & \cdots & 0 & r_1\mu_1 I & A_{N_1} \end{bmatrix}, \tag{3}$$

where r_1 is the total number of repairmen who have the skill to repair machines of type 1, A_i are square matrices of order $\prod_{j=2}^{N}(N_j+1)$, and matrix I is the identity matrix of the same size. The elements of submatrices A_i, $i=1,2,\ldots,N_1$, can be obtained using (2).

To find the steady state probability for each state of the system, let $\mathbf{P}_j$ be the vector of steady state probability corresponding to the vector of state $\mathbf{X}_j$. Then, the balance equations for this system are

$$\begin{aligned} &\mathbf{P}_0 A_0+\mathbf{P}_1\mu_1 I=0,\\ &\mathbf{P}_{j-1}(N_1-j+1)\lambda_1 I+\mathbf{P}_j A_j+\mathbf{P}_{j+1}(j+1)\mu_1 I=0,\\ &\qquad\qquad 1\le j<r_1,\\ &\mathbf{P}_{j-1}(N_1-j+1)\lambda_1 I+\mathbf{P}_j A_j+\mathbf{P}_{j+1}r_1\mu_1 I=0,\quad r_1\le j<N_1,\\ &\mathbf{P}_{N_1-1}\lambda_1 I+\mathbf{P}_{N_1}A_{N_1}=0, \end{aligned}$$

with the normalization condition $\sum_{j=0}^{N_1}\mathbf{P}_j\mathbf{e}=1$, where $\mathbf{e}$ is a column vector of ones.

Due to the quasi-birth-and-death structure of the transition rate matrix **Q**, the steady state probabilities $\mathbf{P}_j$ have the following recursive relation:

$$\mathbf{P}_{N_1}=-\mathbf{P}_{N_1-1}\lambda_1 I[A_{N_1}]^{-1}, \tag{4}$$

$$\mathbf{P}_j=\mathbf{P}_{j-1}\mathbf{C}_j,\quad 1\le j<N_1, \tag{5}$$

where $\mathbf{C}_0=I$, $\mathbf{C}_{N_1}=-\lambda_1 I[A_{N_1}]^{-1}$, and $\mathbf{C}_j$ is as follows:

$$\mathbf{C}_j=\begin{cases} -(N_1-j+1)\lambda_1 I\{A_j+\mathbf{C}_{j+1}(j+1)\mu_1 I\}^{-1}, \\ \qquad\qquad 1\le j<r_1, \\ -(N_1-j+1)\lambda_1 I\{A_j+\mathbf{C}_{j+1}(r_1)\mu_1 I\}^{-1}, \\ \qquad\qquad r_1\le j<N_1. \end{cases} \tag{6}$$

Now, from the balance equations we will have

$$\mathbf{P}_0 A_0+\mathbf{P}_1\mu_1 I=0, \tag{7}$$

which means that $\mathbf{P}_0$ satisfies the following:

$$\mathbf{P}_0\{A_0+\mathbf{C}_1\mu_1 I\}=0. \tag{8}$$

Equation (8) provides a set of $\prod_{j=2}^{N}(N_j+1)$ equations that involve only steady state probabilities corresponding to $\mathbf{P}_0$. Note that $\mathbf{C}_1$ in (8) is known, since it can be obtained recursively using (6). Using the set of Equations (8) and also the normalizing condition (9) as follows:

$$\sum_{j=0}^{N_1}\mathbf{P}_j\mathbf{e}=\mathbf{P}_0\left\{\sum_{j=0}^{N_1}\prod_{i=0}^{j}\mathbf{C}_i\right\}\mathbf{e}=1, \tag{9}$$

we obtain the steady state probability vector $\mathbf{P}_0$. The remaining steady state probability vectors can then be calculated using (5). These steady state probabilities are then used to compute $E[B_1(M)]$, the average number of broken machines of type 1 in the system with skill set matrix M:

$$E[B_1(M)]=\sum_{j=0}^{N_1}j\mathbf{P}_j\mathbf{e}=\mathbf{P}_0\left\{\sum_{j=0}^{N_1}j\prod_{i=0}^{j}\mathbf{C}_i\right\}\mathbf{e}. \tag{10}$$

The average downtime for the machines of type 1 is $E[B_1(M)]/\Lambda_1(M)$ where $\Lambda_1(M)$ is the effective failure rate of type 1 machines under skill set matrix M, and can be obtained as follows:

$$\Lambda_1(M)=\sum_{j=0}^{N_1}(N_1-j)\lambda_1[\mathbf{P}_j\mathbf{e}]=\lambda_1\mathbf{P}_0\sum_{j=0}^{N_1}(N_1-j)\left\{\prod_{i=0}^{j}\mathbf{C}_i\right\}\mathbf{e}.$$

Using the same approach, one can obtain $E[B_i(M)]$ for $i=2,3,\ldots,N$. Therefore, $E[D(M)]$, the total average machine downtime cost per unit time is

$$E[D(M)]=\sum_{i=1}^{N}c_i E[B_i(M)].$$

The expected maximum percentage of broken machines among all machine types in the system with skillset matrix M can be calculated as

$$E[P(M)]=\sum_{\mathbf{x}\in\mathscr{X}}p_{\mathbf{x}}\max_{i\in\{1,2,\ldots,N\}}\frac{x_i}{N_i}. \tag{11}$$

Remark 1. As the above results show, in machine-repairman problems with partially cross-trained repairmen, the skill set matrix M has a significant influence on the average number of broken machines in the system. Therefore, repairman training decisions that affect this matrix are of great importance. In online Appendix C, we present a heuristic approach that uses our queueing model and obtains the optimal or close-to-optimal training decisions that minimize the total average machine downtime and repairman cost. The appendix shows that finding the optimal training decisions is a complex optimization problem; however, myopic heuristic algorithms can be developed to obtain the optimal or close-to-optimal training decisions.

4. Numerical Study

In this section, we report the results of our numerical study. The objectives of the numerical study are (i) to evaluate the effectiveness of our machine-priority and repairman-assignment rules in minimizing the total downtime cost of the system, (ii) to evaluate the effectiveness of our machine-priority and repairman-assignment rules in balancing the percentage of working machines, and (iii) to determine the repair/maintenance environment in which the well-known chain structure cross-training scheme provides the most benefits. Our numerical study provides new insights on design and control issues such as machine-prioritization and repairman-assignment in repair/maintenance environments with partially cross-trained repairmen.

Our numerical study is based on 225 cases that were generated using all possible combinations of scenarios shown in Table 3 for a machine-repairman system that has four types of machines ($N = 4$) and four repairmen ($K = 4$). In Table 3, scenarios for machine downtime costs are chosen such that machines have the same downtime cost, or when costs are different, the difference between c_i and c_{i+1} increases (or decreases) in a linear or nonlinear fashion (i.e., $|c_i - c_{i+1}|$ increases or decreases in a linear or nonlinear fashion). The scenarios for number of machines include cases where different machine types have the same or different number of machines. For scenarios with different numbers of machines, the machine type with the largest number of machines has four times more machines than the machine type with the least number of machines. Scenarios for machine availabilities also include cases in which machines have the same or different availabilities. Scenarios with different machine availabilities are designed to include machines with availabilities ranging from 0.75 to 0.95. Choices for skill set matrices include matrices with a total number of 8, 9, 10, and 12 skills. These skill set matrices include chain (matrix M_1) and pyramid (matrix M_2). The pyramid skill set matrix is representative of a repair crew that includes repairmen with different amounts of experience, i.e., including repairmen with one, two, three, and four skills. For detailed explanation of our numerical experiment, see online Appendix A.

Table 3 Scenarios for Test Cases in Numerical Study

Scenarios for costs		Scenarios for number of machines		Scenarios for machine availabilities	
No.	(c_1, c_2, c_3, c_4)	No.	(N_1, N_2, N_3, N_4)	No.	$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$
1	(1.0, 1.0, 1.0, 1.0)	1	(5, 5, 5, 5)	1	(95%, 90%, 85%, 75%)
2	(0.1, 0.3, 0.6, 1.0)	2	(2, 4, 6, 8)	2	(75%, 85%, 90%, 95%)
3	(1.0, 0.6, 0.3, 0.1)	3	(8, 6, 4, 2)	3	(80%, 80%, 80%, 80%)
4	(0.1, 0.6, 0.8, 0.9)				
5	(0.9, 0.8, 0.6, 0.1)				

4.1. Minimizing Total Average Downtime Cost

As we mentioned in §3, the optimal policy does not follow a simple rule for prioritizing machines. In fact, the priority of machines over each other changes as the number of broken machines in the system changes. It is interesting to investigate how robust the optimal policy is with respect to its dynamic machine-prioritization rule. In other words, if one decides to use a simple static priority policy that does not change with the state of the system (i.e., the number of broken machines of each type), how bad would the performance of this policy be?

All results in §4.1 correspond to the objective of minimizing the total expected machine downtime cost. Section 4.2 presents a similar study for cases where the objective is to balance the percentage of working machines.

4.1.1. The Performance Evaluation of Machine-Priority Rules. In this section, we compare the performance of the optimal policy, which we refer to as the global optimal policy, with the performance of the following machine-priority policies: (i) the optimal $c\mu$ policy, (ii) the optimal $c\mu/r\lambda$ policy, and

(iii) the optimal static priority policy. In order to measure the effectiveness of policy π, $\pi \in \{\text{Optimal } c\mu, \text{Optimal } c\mu/r\lambda, \text{Optimal static}\}$ compared to the global optimal policy in minimizing total average downtime cost, we define Δ_π^D as follows:

$$\Delta_\pi^D = \frac{E[D_\pi^*] - E[D^*]}{E[D^*]} \times 100\%;$$

$$\pi \in \{\text{Optimal } c\mu, \text{Optimal } c\mu/r\lambda, \text{Optimal static}\},$$

where $E[D^*]$ is the total average downtime cost under the global optimal policy, and $E[D_\pi^*]$ is the total average downtime cost under policy π.

Note that although the $c\mu$ rule resolves the issue of which machine has the highest priority for repair, it does not provide any answer to the question of which repairman should perform the next repair (i.e., the repairman-assignment rule). We revised our MDP and obtained $E[D_{c\mu}^*]$, the total average downtime cost under the $c\mu$ rule and its corresponding optimal repairman-assignment rule (i.e., the optimal $c\mu$ policy). We did this by reducing the action space $\mathbf{\Phi}(\mathbf{x})$ in (1) to $\mathbf{\Phi}_{c\mu}(\mathbf{x})$ ($\mathbf{\Phi}_{c\mu}(\mathbf{x}) \subset \mathbf{\Phi}(\mathbf{x})$), which includes only repairman-assignments that follow the $c\mu$ machine-priority rule. This isolates the effects of repairman-assignment rules, and therefore the difference between the performance of the optimal $c\mu$ policy and the global optimal policy will only be the result of using the simple $c\mu$ machine-priority rule instead of the complex machine-priority rule followed by the global optimal policy. Throughout our numerical study, we executed the MDP models such that the optimal solution is 99.9% accurate.

Similar to the optimal $c\mu$ policy, $E[D_{c\mu/r\lambda}^*]$, the total average downtime cost under the optimal $c\mu/r\lambda$ policy can be obtained by reducing the action space $\mathbf{\Phi}(\mathbf{x})$ in (1) to $\mathbf{\Phi}_{c\mu/r\lambda}(\mathbf{x})$ ($\mathbf{\Phi}_{c\mu/r\lambda}(\mathbf{x}) \subset \mathbf{\Phi}(\mathbf{x})$), which includes only repairman-assignments that follow the $c\mu/r\lambda$ priority rule.

The optimal static priority policy is the policy that results in the minimum total average downtime cost of the system while the system follows a static machine-priority rule. For our examples with four machines, there are $4! = 24$ different ways that the machines can be prioritized (with the $c\mu$ and $c\mu/r\lambda$ rules being two of those 24 ways). To obtain the optimal static policy for each case of our numerical study, we took the following two steps: (i) we chose the first static priority rule among the 24 possible rules, and (ii) we used a revised version of our MDP model in (1) to obtain the optimal average downtime cost of the system under the first priority rule. In the revised version of our MDP model, the action space $\mathbf{\Phi}(\mathbf{x})$ is reduced to $\mathbf{\Phi}_{p_1}(\mathbf{x})$ ($\mathbf{\Phi}_{p_1}(\mathbf{x}) \subset \mathbf{\Phi}(\mathbf{x})$) that only includes repairman-assignments that follow the first static priority rule (among the 24). We repeated this process for the remaining 23 possible static priorities, and obtained the minimum total average downtime cost among all 24 cases. We denote the total average downtime cost under the optimal static policy by $E[D_{\mathcal{P}^*}^*]$. Note that the difference between the performance of the optimal static policy and that of the global optimal policy reflects the effects of compromising in system performance by using simple static machine-priority rules instead of the complex machine-priority rule under the global optimal policy.

Based on the 225 cases in our numerical study, we found that $\Delta_{\mathcal{P}^*}^D$ has an average and maximum of 0.05% and 0.98%, respectively. These numbers are 6.55% and 78.74% for $\Delta_{c\mu}^D$, and 1.70% and 16.04% for $\Delta_{c\mu/r\lambda}^D$.

Observation 1. The optimal static priority rule is effective in minimizing total average downtime cost.

The above numbers illustrate the fact that the complex machine-priority rule dictated by the global optimal policy does not have a significant effect in minimizing the total average downtime cost. As $\Delta_{\mathcal{P}^*}^D$ shows, if one decides to implement a simple static priority rule (i.e., the optimal static priority policy) instead of the complex machine-priority scheme of the global optimal policy, one loses less than 1% (at most) in performance.

Remark 2. The optimal static policy is also effective in minimizing the total average number of broken machines in the system. For the 45 cases of our numerical study in which machines have the same downtime cost (i.e., Cost Scenario 1 of Table 3), the average and the maximum $\Delta_{\mathcal{P}^*}^D$ were 0.07% and 0.25%, respectively. Note that in these 45 cases, minimizing total average machine downtime is equivalent to minimizing the total average number of broken machines.

Observation 2. Prioritizing machines according to the $c\mu/r\lambda$ rule results in a close-to-global optimal performance in minimizing total average number of broken machines.

As we mentioned, giving higher priority to machines according to the $c\mu/r\lambda$ rule results in an average of 1.7% and a maximum of 16.04% more cost than that of the global optimal policy. However, for the 45 cases where machines have the same downtime costs, we found that the average and maximum are 0.67% and 2.83%, respectively. This indicates that the $c\mu/r\lambda$ rule (which is equivalent to the $\mu/r\lambda$ rule when all c_is are the same) is an effective policy in minimizing the average number of broken machines in the system.

Although the $c\mu$ rule is proven to be optimal in prioritizing tasks in many production and service environments, it performs poorly in prioritizing machines in our repair/maintenance environment. The $c\mu$ rule, even if used with its corresponding optimal repairman assignment rule, can result in up to 78% more cost than the cost of the global optimal policy. As we mentioned, the reason is that repair/maintenance environments, as opposed to most production and service environments, are finite population queueing systems.

We conclude this section by emphasizing that in maintenance environments with partially cross-trained repairmen, the optimal static machine-priority rule is as good as the complex dynamic machine-priority rules of the global optimal policy. Furthermore, when the objective is to minimize the total average broken machines, static machine-priority rules—such as the $\mu/r\lambda$ rule—which take into account repairmen's skill sets, can perform as well as the complex machine-priority rules dictated by the global optimal policy. Note that we did not evaluate the performance of HPB machine-priority policy with respect to minimizing total expected downtime cost, since the HPB policy is not designed to minimize cost.

4.1.2. The Performance Evaluation of Repairman-Assignment Rules. In §4.1.1, each machine-priority policy was following its corresponding optimal repairman-assignment rule, obtained by solving the revised MDP model. This enabled us to study the effect of machine-priority rules on system performance. In this section, however, we turn our attention to the effects of repairman-assignment rules on the performance of a system. Specifically, assuming the optimal static machine-priority rule ($\mathcal{P}^*$), we compare the performance of the repairman-assignment rules we introduced in §3.3 with the performance of the global optimal policy.

Define $E[D^{\mathcal{R}}_{\mathcal{P}^*}]$ as the total average downtime cost of a system under repairman-assignment rule $\mathcal{R}$ and the optimal static machine-priority rule $\mathcal{P}^*$. Also, define the relative error $\Delta^D_{\mathcal{R}}$ as follows:

$$\Delta^D_{\mathcal{R}} = \frac{E[D^{\mathcal{R}}_{\mathcal{P}^*}] - E[D^*]}{E[D^*]} \times 100\%; \qquad \mathcal{R} \in \{LSR, LVR, LLP, LRR\}.$$

This measures how much the total average downtime cost increases if one decides to implement repairman-assignment rule $\mathcal{R}$ under a static machine-priority rule, rather than following the global optimal policy.

For each of the 225 cases of our numerical study, when $\mathcal{R} \in \{LSR, LVR, LLP, LRR\}$, $E[D^{\mathcal{R}}_{\mathcal{P}^*}]$ was obtained by (i) solving the queueing model presented in §3.4 under $N! = 4! = 24$ machine-priority rules, and then (ii) choosing the system with the minimum total average downtime cost among the 24 cases. In all 24 queueing models, the repairman-assignment rule was $\mathcal{R}$. Based on our numerical study, we found that Δ^D_{LLP} has an average and maximum 0.21% and 2.32%, respectively. These numbers are 2.01% and 8.15% for Δ^D_{LSR}; 1.04% and 7.55% for Δ^D_{LVR}; and 0.82% and 7.65% for Δ^D_{LRR}.

Observation 3. The LLP policy, if implemented under the proper static machine-priority rule, is a very effective repairman-assignment rule in minimizing the total average downtime cost.

Note that the LLP policy assigns repairmen to machines based only on repairman skills and machine downtime costs. This rule does not use machine characteristics (such as failure and repair rates) in the way it assigns repairmen to machines.

Although not as effective as LLP, the LRR rule performs reasonably well in minimizing total average downtime cost. The LRR rule results, *on average*, in less than 1% more expected cost than the global optimal policy. The maximum cost difference we observed was 7%.

We have also compared the performance of the policy that follows the $c\mu/r\lambda$ machine-priority rule and the LLP repairman-assignment rule with the performance of the global optimal policy. We found that, in general, the combination of the $c\mu/r\lambda$ and LLP rules creates a policy that results in an average 1.82% and a maximum of 16.37% more cost than the global optimal cost. However, when minimizing the total average number of broken machine is concerned (i.e., $c_i = c, \forall i$), these numbers reduce to 1.42% and 9.59%, respectively.

We conclude this section by emphasizing that, when minimizing machine downtime costs is of interest in systems in which machines are appropriately prioritized, machine characteristics such as the number of machines and machine availabilities are not essential for developing efficient repairman-assignment rules. Repairman-assignment rules such as the LLP rule, which uses only repairman skill sets and machine downtime costs as the basis for their repairman assignment to machines, can result in a performance that is close to that attained under the global optimal policy.

4.2. Balancing Percentage of Broken Machines

In previous sections, we have shown that the optimal static priority rule and our simple repairman-assignment rules can perform close to the global optimal policy with respect to minimizing the total average downtime cost. In this section, we evaluate the performance of our machine-priority and repairman-assignment rules with respect to its effectiveness in balancing the percentage of working machines (i.e., minimizing the maximum percentage of broken machines among all types).

4.2.1. The Performance Evaluation of Machine-Priority Rules. We examined 45 cases generated by Table 3 (without its scenarios for downtime costs) along with our skill set matrices. Our objective was to study the effectiveness of the following machine-priority policies in balancing the percentage of working machines: (i) policy $\mathcal{P}^*$, i.e., the optimal static priority policy, and (ii) policy HPB^*, i.e., the optimal HPB policy. Note that, as opposed to §4.1, here the optimal static machine-priority rule is the static priority rule that minimizes the maximum percentage of broken machines among different types. We do not evaluate the performance of the optimal $c\mu$ or the optimal $c\mu/r\lambda$ policies here, since minimizing downtime cost is not of interest.

For each case of our numerical study we solved the MDP model (1) with cost function $C(\mathbf{x}) = \max_i\{x_i/N_i\}$ to obtain $E[P^*]$, the expected maximum percentage of broken machines among different types. We evaluated the effectiveness of policy π ($\pi \in \{\mathcal{P}^*, HPB^*\}$), by comparing its performance with the performance of the global optimal policy through Δ_π^P:

$$\Delta_\pi^P = \frac{E[P_\pi^*] - E[P^*]}{E[P^*]} \times 100\%; \quad \pi \in \{\mathcal{P}^*, HPB^*\},$$

where $E[P_\pi^*]$ is the expected maximum percentage of broken machines (among all machine types) under policy π.

The optimal static priority policy is the static priority policy among $N! = 24$ combinations that results in the least expected maximum percentage of broken machines. As explained in the case of minimizing total average downtime cost, the choice of machine-priority rule does not resolve the issue of which repairman should perform the next repair. Therefore, to obtain $E[P_{\mathcal{P}^*}^*]$, the expected maximum percentage broken under the optimal static policy, we used the same approach as in §4.1.1, except we used $C(\mathbf{x}) = \max_i\{x_i/N_i\}$. We have also revised our MDP to obtain $E[P_{HPB^*}^*]$, the expected maximum percentage of broken machines under HPB and its corresponding optimal repairman-assignment rule.

Observation 4. The HPB policy is a very effective machine-priority policy in balancing the percentage of working machines of different types.

When the objective is to balance the percentage of broken machines, in our numerical study we observed that the average and maximum errors of the HPB policy are 0.27% and 0.98%, respectively. Therefore, if one decides to implement the HPB policy instead of the complex machine-priority policy used by the global optimal policy, one does not compromise much on the system's performance.

Note that HPB policy is a myopic policy because it assigns priorities to machines according to the current state of the systems without considering machine characteristics such as machine failure and repair rates. In our numerical study, we observed many cases where the global optimal policy did not assign priorities to machine myopically. Nevertheless, as the above numbers show, the HPB's myopic approach is very effective.

In our numerical study we also observed that the optimal static priority policy results in an average and maximum error 3.8% and 12.9%, respectively. Comparing these numbers with the corresponding numbers in §4.1.1, we can conclude the following:

Observation 5. Static machine-priority policies are more effective in minimizing the system's expected downtime costs than in balancing the percentage of working machines.

4.2.2. The Performance Evaluation of Repairman-Assignment Rules. In this section, we compare the performance of policy $(\mathcal{P}, \mathcal{R})$ that implements machine-priority rule $\mathcal{P}$ ($\mathcal{P} \in \{\mathcal{P}^*, HPB\}$) and follows repairman-assignment rule $\mathcal{R}$ ($\mathcal{R} \in \{LSR, LLP, LRR\}$), with the performance of the global optimal policy that balances the percentage of working machines among all machine types. Policy $(\mathcal{P}^*, \mathcal{R})$ is the policy that implements repairman-assignment rule $\mathcal{R}$ and follows the optimal static machine-priority rule (among 24 possible priority rules) that minimizes the maximum percentage of broken machines among all machine types. On the other hand, policy $(HPB, \mathcal{R})$ is the policy that implements repairman-assignment rule $\mathcal{R}$ and the HPB machine-priority rule. We evaluate the effectiveness of the repairman-assignment rules by the relative error $\Delta^P_{\mathcal{R}}$,

$$\Delta^P_{(\mathcal{P},\mathcal{R})} = \frac{E[P_{(\mathcal{P},\mathcal{R})}] - E[P^*]}{E[P^*]} \times 100\%;$$
$$\mathcal{R} \in \{LSR, LLP, LRR\},\ \mathcal{P} \in \{\mathcal{P}^*, HPB\},$$

where $E[P_{(\mathcal{P},\mathcal{R})}]$ is the expected maximum percentage of broken machines (among different machine types) under policy $(\mathcal{P}, \mathcal{R})$. Note that for dynamic machine-priority policy HPB, (i) we revised our repairman-assignment rules by setting $c_i = 1$ for all machine types, since minimizing cost is not of concern here. When $c_i = 1$ for all machine types, repairman-assignment rules LVR and LSR become the same. (ii) We did not consider the LLP rule, since this rule cannot be used in systems with dynamic machine-priority rules. (iii) We revised the tiebreaking rule in LSR and LRR to a random selection, since the tiebreaking rule was based on static machine-priority rules.

In order to obtain $E[P_{(\mathcal{P}^*,\mathcal{R})}]$, similar to §4.1.1, we used our queueing model. However, to obtain $E[P_{(HPB,\mathcal{R})}]$, we developed a simulation program in C++. The programs are executed for the simulation horizon of 35,000 to 40,000 hours that results in at least 20,000 failures in each machine type. The warm-up period is chosen such that the result data are collected after 5,000 failure events in each type of machines.

From our experiments, we found that the HPB rule results in average and maximum errors around 0.5% and 1.2%, respectively, no matter which repairman-assignment rule among LSR and LRR is followed. This was the same for the optimal static machine-priority policy, which results in an average and maximum errors around 6% and 18%, respectively, regardless of what repairman-assignment rule is implemented. This is different from what we observed under the objective of minimizing the system's downtime cost. As shown in §4.1.2, when the objective is to minimize the total average downtime cost, the maximum error of LSR, LLP, LRR repairman-assignment rules under optimal static machine-priority rules varies from 2.32% to 7.65%. Thus, we conclude the following:

Observation 6. When the goal is to balance the percentage of working machines, the performance of the optimal static machine-priority rule and HPB rule do not significantly change with changes in repairman-assignment rules among $\{LSR, LLP, LRR\}$.

4.3. Chain Skill Set Structure

In repair/maintenance environments in which there are K repairmen and K different machine types (i.e., $N = K$), a two-skill chain structure is a structure in which each repairman can repair two types of machines, each machine type can be served by two repairmen, and the machine types can always be renumbered such that repairman k will have skills to repair machine types k and $k+1$, while repairman K will have the skills to repair machine types K and 1. The following skill set matrix is a two-skill chain matrix in a repair environment with three repairmen and three machine types:

$$M_{chain} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

Many articles in the literature recommend the well-known chain skill set structure as an alternative to a fully cross-trained system in production environments (see Sheikhzadeh et al. 1998, Jordan and Graves 1995, Hopp et al. 2004, and Jordan et al.

2004). For infinite population finite capacity queueing systems, Gurumurthi and Benjaafar (2004) show that chain skill sets are more efficient in symmetric systems, i.e., in queueing systems in which different customer types have the same arrival and service rates. On the other hand, Jordan et al. (2004) show the robustness of the chain skill set structure with respect to system parameters and control decisions in a maintenance system. In this section, we will investigate the performance of the chain structure in finite population queueing systems such as our repair/maintenance environments. More specifically, we would like to find repair/maintenance environments in which chain structures perform very effectively. It is obvious that under complete symmetry (i.e., with the same downtime costs, same failure rates, same repair rates, and same number of machines for all types) the chain will be an efficient structure. However, we would like to investigate whether there exists any asymmetric environment in which the chain is still an efficient structure for the skill sets of repairmen.

Consider the following two repair/maintenance environments with $N = 3$ machine types and $K = 3$ repairmen. In environment E1, $(c_1, c_2, c_3) = (1, 2.5, 5)$, $(N_1, N_2, N_3) = (2, 4, 8)$, $(\lambda_1, \lambda_2, \lambda_3) = (6, 0.4, 10)$, and $(\mu_1, \mu_2, \mu_3) = (20, 15, 40)$; in environment E2, $(c_1, c_2, c_3) = (2, 1, 1.5)$, $(N_1, N_2, N_3) = (6, 8, 2)$, $(\lambda_1, \lambda_2, \lambda_3) = (1, 2, 3)$, and $(\mu_1, \mu_2, \mu_3) = (11, 14, 6)$. As these numbers show, there is a large asymmetry in the cost, numbers of machines, failure rates, and repair rates in both environments. For instance, in E2 the downtime of type 1 machines is twice costlier than that of type 2 machines, type 2 machines have four times more machines compared to type 3 machines, the failure rate of type 3 machines is three times larger than that of type 1 machines, and the repair rate of type 2 machines is more than twice larger than that of type 3 machines. The question that we focus on here is whether a chain structure is a good fit for the repairman skill set in E1 and E2.

To investigate this, we assume that the total number of skills in the system is six, so that a two-skill chain structure is feasible for this environment. Next, we investigate whether there exists any other skill set structure (with a total of six skills) that performs better than a two-skill chain structure in the above asymmetric environment. We developed a computer program and generated all possible skill set matrices with a total of six skills. The program resulted in 16 different matrices (including the two-skill chain structure).

For each of 16 matrices, we solved our MDP model and obtained the optimal total average downtime cost when each matrix is used in the above environment. Then, we chose the skill set matrix among the 16 that results in the minimum average downtime cost. We call that the optimal skill set matrix.

For E1 we found that the two-skill chain results in 21% higher cost than that under the optimal skill set matrix. This is not unexpected, since E1 is an asymmetric environment. However, to our surprise, we found that in E2, which is also an asymmetric environment, the two-skill chain matrix M_{chain} is indeed the optimal skill set matrix.[3] This motivated us to further explore the differences between the two environments. After studying several other examples, we found that, although there is not an apparent symmetry in the parameters (i.e., c_i, N_i, λ_i, and μ_i) in E2, the environment is indeed symmetric with respect to $c_i N_i (1-\alpha_i)$, where α_i is the availability of machines of type i. For example, in E2, we have $c_i N_i (1-\alpha_i) = 1$ for all machine types ($i = 1, 2, 3$). We call the symmetry in $c_i N_i (1-\alpha_i)$, the hidden symmetry.

In order to further explore this observation, we examined 108 more cases with perfect hidden symmetry generated through two different approaches, namely a systematic approach and a random approach. For the details of how these 108 cases were generated, see the online Appendix B.

For each case generated by our systematic or random approaches, we used our MDP to calculate the optimal average downtime cost under each of 16 possible skill set matrices, and we obtained the minimum among those 16 optimal costs. We call that $E[D(opt.\ skill)]$. For each case, we also obtained $E[D(chain)]$, the optimal downtime cost under the chain skill set. We compared the optimal cost under the chain skill set with the

[3] The range for the costs of the 16 skill set matrices was large. In fact, one of the possible 16 matrices resulted in the total average downtime cost that was 34.02% more than the cost of the chain skill set structure.

Table 4 Performance of Chain Skill Set Structure

	Number of cases
(Chain is optimal) $\Delta_{Chain} = 0$	79 out of 108 (i.e., 73% of cases)
$\Delta_{Chain} < 1\%$	92 out of 108 (i.e., 85% of cases)
$\Delta_{Chain} < 2\%$	98 out of 108 (i.e., 91% of cases)
$\Delta_{Chain} \geq 2\%$	10 out of 108 (i.e., 9% of cases)
Ave. Δ_{Chain}	1.29%
Max Δ_{Chain}	2.49%

optimal cost under the optimal skill set through the following measure:

$$\Delta_{Chain} = \frac{E[D(chain)] - E[D(opt.\ skill)]}{E[D(opt.\ skill)]} \times 100\%.$$

This measure shows how close the performance of chain structure is to the performance of the optimal skill set structure. The results for all 108 cases are summarized in Table 4.

As the table shows, in environments that are symmetric with respect to $c_i N_i(1-\alpha_i)$, even when the chain structure is not the best structure among the 16 structures, its performance is nevertheless very close to that of the optimal structure. In fact, the average percentage difference in our 108 examples was less than 1.29%. We would like to emphasize that, in almost all cases of our numerical study, the difference in the cost of 16 skill set matrices was very significant and was sometimes up to 45%. The small relative error Δ_{Chain}, is therefore a good indication of the effectiveness of the chain skill set in systems with hidden symmetry.

The intuition behind the good performance of the chain structure in environments with symmetry in $c_i N_i(1-\alpha_i)$ can be described as follows. Consider a system with unlimited repairmen. In that system, $N_i(1-\alpha_i)$ is the average number of broken machines of type i in the system. Therefore, $c_i N_i(1-\alpha_i)$ can be considered as a rough estimate for the average downtime cost incurred per unit time by type i. Having symmetry in $c_i N_i(1-\alpha_i)$ indicates that all machine types generate almost the same downtime cost, and thus require the same level of attention. This phenomenon can also be observed in environments with complete symmetry. Thus, a chain structure, which is one of the most effective symmetric structures, will perform well in this environment.

Note that symmetry in a chain structure is not the only reason for its good performance. The particular connectivity in its structure that allows more ways of shifting repairman capacities in the system makes it more effective than other symmetric structures. (See Iravani et al. 2004 for structural properties of chain skill sets.)

5. Conclusion and Further Research

The introduction of a wide range of new technologies that often include complex machinery has resulted in repair environments in which the repairmen are not fully cross-trained. This paper has focused on several issues in these environments including the strategy of assigning repairmen to machines and prioritization of machines for repair. We considered two different objectives, namely minimizing the total average downtime cost of the system and balancing the percentage of working machines across different types of machines. We have shown that, although the global optimal policy has a complex structure, using simple machine-priority and repairman-assignment rules can yield as good a performance as the global optimal policy. Finally, we studied environments in which the well-known chain skill set structures perform better than others. We found that those environments share a (hidden) symmetry that is more general than an apparently complete symmetry.

Future research should focus on systems in which repair operations cannot be preempted, or systems in which repairmen work at different speeds in repairing different types of machines. This will also model systems in which repairmen have different levels of experience. Also, it would be interesting to explore the effect of partially cross-trained repairmen in systems where machines have three states, namely, (i) working, (ii) deteriorated (thus producing defective items), and (iii) broken. Another avenue for future research is the study of systems with nonlinear cost functions. For those cases, static priority rules may not be effective, and some extension of the generalized $c\mu$ rule for these closed queueing systems would be of great interest.

Acknowledgments
This work was supported in part by NSF under Grant DMI-0000125.

References

Agnihothri, S. R. 1989. Interrelationships between performance measures for the machine-repairman problem. *Naval Res. Logist.* **36** 265–271.

Bahnasawi, A. A., M. S. Mahmoud, S. Z. Eid. 1996. Sensitivity analysis of machine interference in manufacturing systems. *Comput. Indust. Engrg.* **30** 753–764.

Chandra, M. J. 1986. A study of multiple finite-source queueing models. *J. Oper. Res. Soc.* **37** 275–283.

Chandra, M. J., R. G. Sargent. 1983. A numerical method to obtain the equilibrium results for the multiple finite source priority queueing model. *Management Sci.* **29** 1298–1308.

Cox, D. R., W. L. Smith. 1961. *Queues*. Methuen, London, UK; Wiley, NY.

Gans, N., G. Koole, A. Mandelbaum. 2003. Telephone call centers: Tutorial, review and research prospects. *Manufacturing Service Oper. Management* **5** 79–141.

Gurumurthi, S., S. Benjaafar. 2004. Modelling and analysis of flexible queueing systems. *Naval Res. Logistics* **51** 755–782.

Hopp, W. J., M. P. Van Oyen. 2004. Agile workforce evaluation: A framework for cross-training and coordination. *IIE Trans.* **36** 919–940.

Hopp, W. J., E. Tekin, M. P. Van Oyen. 2004. Benefits of skill chaining in production lines with cross-trained workers. *Management Sci.* **50** 83–98.

Iravani, S. M. R., B. Kolfal. 2005. When does the $c\mu$ rule apply to finite-population queueing systems? *Oper. Res. Lett.* **33** 301–304.

Iravani, S. M. R., V. Krishnamurthy, G. H. Chao. 2006. Optimal server scheduling in nonpreemptive finite-population queueing systems. *Queueing Systems*. Forthcoming.

Iravani, S. M., K. Sims, M. P. Van Oyen. 2005. Structural flexibility: A new perspective of manufacturing and service operations. *Management Sci.* **51** 151–166.

Jordan, W. C., S. C. Graves. 1995. Principles on the benefits of manufacturing process flexibility. *Management Sci.* **41** 577–594.

Jordan, W. C., R. R. Inman, D. E. Blumenfeld. 2004. Chained cross-training of workers for robust performance. *IIE Trans.* **36** 953–967.

Kameda, H. 1982. A finite-source queue with different customers. *J. Assoc. Comput. Machinery* **29** 478–491.

Kececioglu, D. 1991. *Reliability Engineering Handbook*. Prentice Hall, Upper Saddle River, NJ.

Koulamas, C. P. 1992. A bicriterian algorithm for minimizing machine interference and reducing job waiting time. *Internat. J. Systems Sci.* **23** 1229–1235.

Koulamas, C. P. 1996. Scheduling two parallel semiautomatic machines to minimize machine interference. *Comput. Oper. Res.* **23** 945–956.

Lippman, S. A. 1975. Applying a new device in the optimization of exponential queueing systems. *Oper. Res.* **23** 687–710.

Mandelbaum, A., A. Stolyar. 2004. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cmu-rule. *Oper. Res.* **52** 836-855.

Pinker, E. J., R. A. Shumsky. 2000. The efficiency-quality trade-off of cross-trained workers. *Manufacturing Service Oper. Management* **2** 32–42.

Shawky, A. I. 1997. The single-server machine interference model with balking, reneging and an additional server for longer queues. *Microelectronics Reliability* **37** 355–357.

Sheikhzadeh, M., S. Benjaafar, D. Gupta. 1998. Machine sharing in manufacturing systems: Total flexibility versus chaining. *Internat. J. Flexible Manufacturing Systems* **10** 351–378.

Stecke, K. E. 1992. Machine interference: Assignment of machines to operators. *Handbook of Industrial Engineering*, 2nd ed., 1460–1494.

Stecke, K. E., J. E. Aronson. 1985. Review of operator machine interference models. *Internat. J. Production Res.* **23** 129–151.

Van Mieghem, J. A. 1995. Dynamic scheduling with convex delay costs: The generalized $c\mu$ rule. *Ann. Appl. Probab.* **5** 808–833.