Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 3 Issue 8, August - 2016

Dispatching a Traveling Repairman to Different Locations with Unreliable Machines

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 the problem of dispatching a single repairman to different locations with unreliable machines. The system consists of N locations and a single repair depot denoted by 0. Location n (for $n \in \{1, \ldots, N\}$) has m_n identical machines which independently and randomly fail during operation, where the failure times follow the exponential distribution with rate λ_n. A downtime penalty h_n is charged per time unit per failed machine in location n. Failures are communicated in real-time to the traveling repairman who restores the failed machines into "as-good-as-new" condition. The repair time S_n of a single machine at location n as well as the travel time $T_{n,v}$ from n to v (for $n, v \in \{1, \ldots, N\} \cup \{0\}$) are stochastic and follow known, but arbitrary, distributions. The objective is to dispatch the repairman so as to minimize the overall downtime. The repairman operations are modeled as a multiclass finite population queueing system. We obtain the optimal dispatching policy by solving an infinite-horizon semi-Markov decision process problem under the expected long-run average cost criterion. Numerical examples are provided to illustrate our approach and to get insights into the dispatching problem.**

Keywords—Traveling repairman; unreliable machines; downtime; optimal dispatching; finite population queues; Markov decision processes

I. Introduction

This paper considers a single repairman who provides on-site corrective maintenance at different locations. Each location corresponds to a production facility using multiple machines that randomly fail during operation. Failed machines incur a downtime penalty due to loss of production or reduced quality of service. We address the basic research question how to dispatch the repairman so as to minimize downtime penalties.

The repairman dispatching problem is particularly important when downtime penalties are large. This is the case of virtually all capital-intensive industries, e.g. energy, mining, telecom, and so on. For instance, an out-of-service wind turbine costs about $7,500 per week due to lost electricity production as reported by a wind farm operator (www.moventas.com). More extreme penalty figures have been reported in other industries such as semiconductor manufacturing. In order to cost-effectively dispatch the repairman, one must deal with nontrivial trade-offs. E.g., should the repairman visit a nearby low-priority location first or visit a distant high-priority location first? Starting with the nearby location would save travel time but at the expense of delaying the high-priority location visit, and vice versa. This kind of trade-off depends in a complicated way on the system parameters, e.g. machines' reliability, travel distances, maintenance requirements, downtime severity, etc. In other words, intuition may not lead to optimal dispatching, and so a computational model is needed.

Several researchers in the "vehicle routing problem" community studied the problem of delivering on-site maintenance services. Ref. [1-3] considered the problem of routing maintenance technicians to satisfy a priori known on-site service requests. Because service requests arrive randomly over time, these models do not capture the queueing delays, which are a major concern when serving delay-sensitive customers such as capital-intensive industries (delay corresponds to the time difference between the arrival and fulfilment of a given request, i.e. downtime in maintenance applications). A joint maintenance-routing problem where delays are calculated was studied in [4]. In addition to being limited to a single machine in each customer location, the model in [4] is static as it does not adapt to new machine failures. There have been limited attempts to incorporate queueing delays into dynamic vehicle routing models, as evidenced by the recent survey by [5]. Ref. [6] analyzed the situation where on-site service requests arrive as a Poisson process to a bounded area. The model in [6] is mostly suitable when the repairman serves a very large number of locations, which may not be the case in applications where the repairman provides on-site service to a moderately large number of locations.

The above vehicle routing models have been primarily motivated by applications with "time window" and "tour duration" constraints, e.g. when the repair service is provided to households. These constraints may not be critical in capital-intensive applications, which motivate our research. Indeed, production sites can be accessed virtually any time during the day (they operate 24/7) alleviating the time window constraints. Furthermore, it is not unusual that a maintenance crew spends several days before returning to the repair depot. This is especially the case when the production sites are remote. E.g., special-purpose vessels, that can accommodate a maintenance crew for a long period of time, are used for the maintenance of offshore platforms. The tour duration constraint can be safely ignored in a similar situation.

Conceptually, our proposed dispatching model is a "polling system". A polling system consists of a single server that attends to multiple parallel queues, where changeover times are incurred when the server switches between queues. Applications of polling systems can be found in production, transportation, and telecommunication [7]. The majority of the published polling models assume that service requests arrive to each queue according to a Poisson process. In other words, requests are generated by infinite calling populations. This assumption can be problematic in our context where there can be as few as a single machine at a particular production site.

We model the repairman operations as a "single server multiclass finite source queueing system". Within this framework, each finite source corresponds to a production site having a finite population of machines, and the server corresponds to the traveling repairman who can repair one machine at a time. Thus, broken machines form parallel queues, one queue per production site, and we wish to schedule the service of these queues by the repairman. This scheduling problem is solved using the Markov decision process approach.

II. Problem Description

A single traveling repairman provides on-site repair to N geographically distributed production sites (i.e. the customers). Denote the customer set by $\mathcal{N} = \{1, \dots, N\}$ and the repair depot by 0. Customer $n \in \mathcal{N}$ owns m_n identical machines which independently break down during opertaion. The time-to-failure of these machines is exponentially distributed with rate λ_n. Failed machines must be taken out-of-service. A downtime penalty h_n is charged per failed customer n machine per time unit. Failure events are communicated in real-time to the traveling repairman who fixes the machines. Let S_n be the stochastic service time required to repair a single broken customer n machine. Also, let $T_{n,v}$ be the stochastic travel time between locations n and v, where $n, v \in \mathcal{N} \cup \{0\}$. All repair and travel times are assumed to follow known distributions. It is assumed that each machine resumes normal operation as soon as it is repaired, i.e. the repaired machine becomes "as good as new". In addition, the repairman must return to the depot when there are no failures in the system.

The objective is to obtain a dynamic dispatching policy for the repairman so as to minimize the expected long-run average downtime cost. The problem just described is modeled as a "single server multiclass finite-source queueing system". Actually, as machines break down, they form repair queues in their respective customer locations, which the repairman (i.e. the server) must visit and serve. We restrict our attention to dispatching policies that are:

- Nonpreemptive, i.e. the repairman cannot be interrupted when he is en-route or when he is repairing a particular machine.
- Nonidling, i.e. the repairman cannot deliberately idle while repair request(s) are pending.
- Nonanticipating, i.e. the repairman cannot travel to a location with an empty repair queue in anticipation of future failure(s) at that location.
- Exhaustive, i.e. the repairman must repair all broken machines in location n (for some $n \in \mathcal{N}$) before he leaves this location.

Let $Q_n(t)$ be the customer n queue length at time t. Obviously, $Q_n(t)$ takes values in the set $\mathcal{Q}_n = \{0,1, \dots, m_n\}$. The queue length vector $(Q_1(t), \dots, Q_N(t))$ at time t takes values in the set $\mathcal{Q} = \mathcal{Q}_1 \times \dots \times \mathcal{Q}_N$. So, the expected long-run average downtime cost associated with some dispatching policy π can be defined as,

$$g(\pi) = \lim_{t\to\infty} \mathbb{E}_\pi \left\{ \int_0^t \sum_{n\in\mathcal{N}} Q_n(x) h_n dx \right\} / t \qquad (1)$$

The expectation in (1) is over the vector-valued stochastic process $Q = \{Q_1(x), \dots, Q_N(x)\}_{x\geq 0}$ given that policy π is followed by the repairman.

III. Optimal Dispatching

We find the optimal policy $\pi^* = \mathrm{argmin}_\pi\{g(\pi)\}$ using the Markov decision process approach. Under this approach, the stochastic process Q is controlled at specific points in time called "decision epochs". The key idea is to choose these epochs in such a way that it is reasonable to make a decision at these epochs and the evolution of Q can be analyzed between any two consecutive epochs. Specifically, we must be able to derive the expected sojourn times, the expected costs, and the transition probabilities (see [8] for more details). Once these elements are specified, the resulting optimality equations can be solved using standard procedures.

Notation. Probability and expectation are denoted $\mathbb{P}\{\cdot\}$ and $\mathbb{E}\{\cdot\}$. Let $\mathbb{I}\{\cdot\}$ be the indicator function, i.e. $\mathbb{I}\{\gamma\} = 1$ if some condition γ is satisfied and $\mathbb{I}\{\gamma\} = 0$ otherwise. Let $\mathbb{B}\{j; k; p\} = \binom{k}{j} p^j (1-p)^{k-j}$, i.e. the probability mass function of a binomial $(k; p)$ random variable. Define e_n as the N-dimensional vector with value 1 in the entry n and values 0 elsewhere. The expectation of S_n is denoted $\bar{S}_n$, i.e. $\bar{S}_n = \mathbb{E}\{S_n\}$. The Laplace-Stieltjes transform of S_n is denoted $\tilde{S}_n(\cdot)$, i.e. $\tilde{S}_n(\lambda) = \mathbb{E}\{\exp(-\lambda S_n)\}$ for some real parameter λ. The same notation is used for travel times $T_{n,v}$.

Decision Epochs. We select the main decision epochs as the points in time when the repairman should make a decision which customer location to visit next, i.e. the dispatching decision. To make the problem more tractable, we introduce also the additional decision epochs which occur whenever the repairman completes the repair of a broken machine at customer location n (for any $n \in \mathcal{N}$) or he arrives at the new location (including the depot). Thus, the decision epochs can be summarized as follows:

i. The first failure occurs while the repairman is idle at the repair depot and the repairman is sent to that location.

ii. Repair completion of a single machine at any customer location $n \in \mathcal{N}$ considering exhaustive service policy.

iii. Travel completion between any two locations $n, v \in \mathcal{N} \cup \{0\}$ such that $n \neq v$, i.e. the decision epoch is the arrival time to the new location.

It follows by the "as good as new" assumption and the "memoryless" property of the exponentially distributed failure times that dispatching decisions rely on two pieces of information. First, the current location of the repairman. Second, the number of failed machines in each customer location. Therefore, the tuple (l, q) is an appropriate "state" representation of the queueing system at any decision epoch, where $l \in \mathcal{N} \cup \{0\}$ is the current location of the repairman and q is the N-dimensional queue length vector whose nth element, $q_n \in \mathcal{Q}_n$, is the number of failed machines in location $n \in \mathcal{N}$. Let $\mathcal{Z} = \{(l, q): l \in \mathcal{N} \cup \{0\}, q \in \mathcal{Q}\}$ denote the state space of the queueing system.

We refer to the decision that dispatches the repairman to location a (including his current location l) as "action" a. If $a \neq l$, then action a means travel to location $a \in \mathcal{N} \cup \{0\}$. Otherwise, it means to remain in the current location $a = l \in \mathcal{N} \cup \{0\}$ until the next decision epoch occurs. The action space is simply $\mathcal{A} = \mathcal{N} \cup \{0\}$. Denote by $\mathcal{A}_{l,q} \subset \mathcal{A}$ the set of permissible actions in state $(l, q) \in \mathcal{Z}$. Given our choice of decision epochs, it is seen that,

$$\begin{aligned} &\mathcal{A}_{l,q} = \{0\}, \quad \forall l \in \mathcal{N} \cup \{0\} \text{ and } q = 0 \\ &\mathcal{A}_{l,q} = \{l\}, \quad \forall l \in \mathcal{N} \text{ and } \forall q \in \mathcal{Q} \text{ s.t. } q_l > 0 \\ &\mathcal{A}_{l,q} = \{n: q_n > 0 \text{ and } n \in \mathcal{N}\}, \\ &\qquad \forall l \in \mathcal{N} \cup \{0\} \text{ and } \forall q \\ &\qquad \in \mathcal{Q} \text{ s.t. } (q \neq 0 \text{ and } q_l = 0) \end{aligned} \tag{2}$$

Here, the 1st line of (2) corresponds to two situations. First, idling the repairman when $q = 0$ and $l = 0$, i.e. the repair queues are empty upon the repairman's arrival to the depot (this is the only option because the policy is nonanticipating). Second, returning to the depot when $q = 0$ upon repair completion of the last failed machine at location l (for some $l \in \mathcal{N}$). The 2nd line of (2) corresponds to initiating a new repair operation at location $l \in \mathcal{N}$ where the repairman currently is (this is the only option because the policy is exhaustive). The 3rd line of (2) corresponds to dispatching the repairman to some location $n \in \mathcal{N}$ ($n \neq l$) where at least one machine has failed $q_n > 0$ (because the policy is nonidling).

Table 1 shows the simulated states at ten consecutive decision epochs of an arbitrary system with three customers ($N = 3$) under an arbitrary dispatching policy. It should be emphasized that some decision epochs in Table 1 are "redundant". E.g., waiting in depot is the only option for the repairman at the 1st decision epoch due to the nonanticipating nature of the dispatching policy. Likewise, traveling to location 2 is the only option at the 2nd decision epoch due to the nonidling nature of the policy. In fact, all decision epochs in Table 1 are redundant except the 4th and 10th epochs. The redundant decision epochs are "artificially" introduced in order to facilitate the calculation of the expected sojourn times, expected costs, and transition probabilities (i.e. the building blocks of the Markov decision model).

Table 1. Illustration of a system with three customer locations ($N = 3$).

Decision epoch	*State*	*Possible actions*	*Action taken*
1	$l = 0$, $q = (0,0,0)$	$\{0\}$	0
	The repairman has returned to the depot and finds that all repair queues are empty. He must wait in the depot until the next failure.		
2	$l = 0$, $q = (0,1,0)$	$\{2\}$	2
	The first failure has occurred at location 2 which the repairman must immediately visit.		
3	$l = 2$, $q = (0,1,1)$	$\{2\}$	2
	The repairman has arrived to location 2. A new failure has also occurred at location 3. The repairman must repair the machine in his current location 2.		
4	$l = 2$, $q = (2,0,1)$	$\{1,3\}$	1
	The repairman has repaired the machine in location 2. Two new failures have occurred at location 1. The repairman can either travel to location 1 or 3. The dispatching policy selects location 1.		
5	$l = 1$, $q = (2,0,1)$	$\{1\}$	1
	The repairman has arrived to location 1 where he must start repairing one of the two failed machines in this location.		
6	$l = 1$, $q = (1,0,1)$	$\{1\}$	1
	The repairman has completed the first repair and must start the second repair.		
7	$l = 1$, $q = (0,0,1)$	$\{3\}$	3
	The repairman has finished repairing the second machine at his current location 1 and must travel to location 3.		
8	$l = 3$, $q = (0,0,1)$	$\{3\}$	3

	The repairman has arrived at location 3 and must start the repair.		
9	$l = 3,\, q = (0,0,0)$	$\{0\}$	0
	The repairman has completed the repair at his current location 3. He must return to depot because all repair queues are empty.		
10	$l = 0,\, q = (1,3,1)$	$\{1,2,3\}$	1
	The repairman has arrived to depot and finds that new failures have occurred while he was en-route from location 3 to the depot. He can visit any of the three customer locations. The dispatching policy selects location 1.		

Expected sojourn times. Let $d_{l,q}(a)$ be the expected sojourn time, i.e. the expected time to the occurrence of the next decision epoch, given that the state is currently (l,q) and action a is taken.

i. When the repair queues are empty at the time the repairman returns to depot (i.e. $l=0, q=0$), then the repairman must remain idle in the depot (i.e. $a=0$) and the next decision epoch will be triggered by the next failure occurrence. In other words, the time to the next decision epoch is the minimum of the failure times of the currently $\sum_{n\in\mathcal{N}} m_n$ operating machines. Because the failure times follow the exponential distribution with rate λ_n, the minimum of the failure times follows the exponential distribution with rate $\sum_{n\in\mathcal{N}} m_n\lambda_n$. Hence, the expected sojourn time in this case is equal to,

$$d_{l,q}(a) = 1/\textstyle\sum_{n\in\mathcal{N}} m_n\lambda_n \qquad (3)$$

ii. When there is at least one failed machine in the repairman's current location $l \in \mathcal{N}$, then the repairman must initiate a new repair at his current location (i.e. $a = l$) and the next decision epoch will occur upon repair completion. Hence, the expected sojourn time in this case is given by,

$$d_{l,q}(a) = \bar{S}_l \qquad (4)$$

iii. When the repairman completes the repair of the last failed machine at his current location $l \in \mathcal{N}$, then he will be dispatched to some customer location $a \neq l$ for which $q_a > 0$ or he will return to depot if $q = 0$. Also, when a failure occurs at customer location $a \in \mathcal{N}$ when the repairman idles in the depot (i.e. $l = 0, q = e_a$), then the repairman will be dispatched to this customer location. In these cases, the next decision epoch occurs upon travel completion. So, the expected sojourn time is equal to,

$$d_{l,q}(a) = \bar{T}_{l,a} \qquad (5)$$

Expected costs between two decision epochs. Let $c_{l,q}(a)$ be the expected downtime penalty incurred between two consecutive decision epochs given that the queueing system is in state (l,q) and action a is taken.

i. Clearly, no downtime is incurred while the repairman idles in the depot because all machines are operating,

$$c_{l,q}(a) = 0 \qquad (6)$$

ii. Consider the cases where the repairman initiates a repair at his current location l. By the time the repair finishes, two types of downtime penalty would occur. First, a direct penalty due to the currently q_n failed machines, $\forall n \in \mathcal{N}$. The direct penalty is simply $\sum_{n\in\mathcal{N}} q_n h_n \bar{S}_l$. Second, a potential penalty due the currently $m_n - q_n$ operating machines, $\forall n \in \mathcal{N}$. In fact, the operating machines could possibly fail while the repair is in-progress. Define $\tau_n(l,q,a)$ as the expected amount of time that a currently operating customer n machine would spend in failed condition while the repair is in-progress. So, the potential penalty cost is given by $\sum_{n\in\mathcal{N}} (m_n - q_n)\tau_n(l,q,a)h_n$. It is shown in the Appendix that $\tau_n(l,q,a) = \bar{S}_l - (1 - \tilde{S}_l(\lambda_n))/\lambda_n$. Adding the direct and potential penalty components yields,

$$c_{l,q}(a) = \textstyle\sum_{n\in\mathcal{N}} \big[m_n\bar{S}_l - (m_n - q_n)(1 - \tilde{S}_l(\lambda_n))/\lambda_n\big]h_n \qquad (7)$$

iii. Consider the cases where the repairman moves to a different location $a \neq l$. By a similar reasoning, the expected downtime cost to be charged by the time of travel completion is equal to,

$$c_{l,q}(a) = \textstyle\sum_{n\in\mathcal{N}} \big[m_n\bar{T}_{l,a} - (m_n - q_n)(1 - \tilde{T}_{l,a}(\lambda_n))/\lambda_n\big]h_n \qquad (8)$$

Transition probabilities. If action a is taken at the current state (l,q), then the state of the queueing system at the next decision epoch will be (a,q') with probability $P(q'|l,q,a)$, i.e. the transition probability. Observe that the state to which the queueing system transitions only depends on the new machine failures (if any) that occur between two consecutive epochs. This is because the repairman's location at the next decision epoch is known with certainty. So, $P(q'|l,q,a)$ is actually the probability that the queue length vector becomes q' at the next decision epoch given that action a is taken at state (l,q).

i. Suppose the repair queues are empty at the time the repairman returns to the depot (i.e. $l = 0, q = 0$) so that the repairman must idle at the depot (i.e. $a = 0$). The state at the next decision epoch will be $(a,q')=(0,e_v)$ if a customer $v\in\mathcal{N}$ machine is the first to fail. Since failure times are exponentially distributed, then the probability of this event is $m_v\lambda_v/\sum_{n\in\mathcal{N}}m_n\lambda_n$. We have,

$$P(q'|l,q,a)=m_v\lambda_v/\sum_{n\in\mathcal{N}}m_n\lambda_n, \quad \forall v\in\mathcal{N} \text{ and } q'=e_v \tag{9}$$

ii. Suppose the repairman initiates a repair at his current location l. A transition into state (a,q') implies that q'_n-q_n machines (out of the currently m_n-q_n operating machines) have joined the repair queue at customer location n by the repair completion time if this customer is not served at the current decision epoch. If customer n is served, then it is implied that q'_n-q_n+1 machines (out of the currently m_n-q_n operating machines) have joined the queue. This is because the customer n machine that will have been repaired by the next decision epoch will immediately leave the repair queue to start operating again. The number of such new failures, for each customer, is a binomial random variable. This assertion follows from the fact that machines fail independently of one another, and that all the operating machines of the same customer have the same probability of failure before the repair terminates. Denote such failure probability by $\rho_n(l,q,a)$. It is shown in the Appendix that $\rho_n(l,q,a)=1-\tilde{S}_l(\lambda_n)$. Hence, the binomial random variable has parameters $(m_n-q_n;1-\tilde{S}_l(\lambda_n))$. In view of the above discussion along with the independent failure occurrences across different customers, we have,

$$P(q'|l,q,a)=\prod_{n\in\mathcal{N}}\mathbb{B}\{q'_n-q_n+\mathbb{I}\{n=l\};m_n-q_n;1-\tilde{S}_l(\lambda_n)\}, \quad \forall q'\in\mathcal{Q}\text{ s.t. }q'\ge q-e_l \tag{10}$$

iii. The transition probabilities when the repairman is dispatched to a different location can be derived in the same way,

$$P(q'|l,q,a)=\prod_{n\in\mathcal{N}}\mathbb{B}\{q'_n-q_n;m_n-q_n;1-\tilde{T}_{l,a}(\lambda_n)\}, \quad \forall q'\in\mathcal{Q}\text{ s.t. }q'\ge q \tag{11}$$

Optimality equations. A stationary dispatching policy is defined as the mapping $\pi:\mathcal{Z}\mapsto\mathcal{A}$ which prescribes action $a\in\mathcal{A}_{l,q}$ whenever the state of the queueing system is found to be $(l,q)\in\mathcal{Z}$ at a given decision epoch. The optimal dispatching policy $\pi^*=\mathrm{argmin}_\pi\{g(\pi)\}$ can be obtained using the value iteration algorithm for semi-Markov decision processes [8]. The optimality equation, $\forall(l,q)\in\mathcal{Z}$, at iteration $n\ge 1$ of the value iteration algorithm is as follows,

$$V_n(l,q)=\min_{a\in\mathcal{A}_{l,q}}\left\{\frac{c_{l,q}(a)}{d_{l,q}(a)}+\left(1-\frac{d}{d_{l,q}(a)}\right)V_{n-1}(l,q)+\frac{d}{d_{l,q}(a)}\sum_{q'\in\mathcal{Q}}P(q'|l,q,a)V_{n-1}(a,q')\right\} \tag{12}$$

The parameter d in (12) can be chosen arbitrarily, where $0<d\le\min_{(l,q)\in\mathcal{Z},a\in\mathcal{A}_{l,q}}\{d_{l,q}(a)\}$.

The optimality equation (12) can be solved iteratively. However, the value iteration algorithm only works for systems with a moderate number of customers and a moderate number of machines in each location. In fact, the proposed semi-Markov decision process formulation quickly suffers from the "curse of dimensionality" due to the multidimensional state space $\mathcal{Z}$.

IV. Numerical Illustration

In this section, we provide a numerical example illustrating our approach to the problem of dispatching a traveling repairman.

Consider a system with four customer locations ($N=4$). The number of machines in each location is $m_1=6,m_2=3,m_3=5,m_4=8$. Further, suppose that all customers use identical machines. In particular, the machines have the same failure $\lambda_n=0.005$ ($\forall n\in\mathcal{N}$), i.e. the mean time between failures is equal to $1/\lambda_n=200$ time units. Also, the machines have the same repair time which is uniform between 6 and 12 time units, i.e. $S_n\sim U(6{,}12)$. However, the customers have different downtime penalties, $c_1=4,c_2=3,c_3=2,c_4=1$ in \$ per time unit. In other words, customers are ordered by decreasing priority level. The travel times are chosen to be deterministic and symetric, where the travel time matrix $T=(T_{n,v})$ is equal to,

$T=$	**0**	**1**	**2**	**3**	**4**
0	0	16	12	8	6
1	–	0	12	20	14
2	–	–	0	8	9
3	–	–	–	0	6
4	–	–	–	–	0

We apply the value iteration algorithm to solve for the optimal policy π^*. The stopping parameter of the algorithm is set to 5%, i.e. the algorithm terminates when the relative difference between the lower and upper bounds for the minimum expected long-run average downtime cost, g, is less than or equal 5%.

The value iteration algorithm converged in about 75 seconds for the considered problem. The minimum expected long-run average downtime cost was found to be \$13.47 per time unit (this value represents the midpoint between the lower bound and upper bound generated by the algorithm).

The value iteration algorithm results reveal an interesting property of the dispatching policy. Namely, the optimal policy is not necessarily monotone in the queue lengths. E.g., Figure 1 shows that it is optimal to dispatch the repairman from his current location $l=1$ to location $a=4$ for all $q=(0,0,2,q_4)$ such that $1 \le q_4 \le 5$. However, when $q_4 \ge 6$ the repairman is dispatched to location $a=3$ (the same phenomenon is depicted in Figure 2). This phenomenon is somewhat counterintuitive because one would expect that the larger the number of failures in a particular location the more desirable it is to visit that location (if all else being equal). A possible explanation for such phenomenon is that the optimality equations (12) chooses not to dispatch the repairman to location $a=4$ when the queue length q_4 exceeds 5 failures because such equations recognize that this action would cause an excessive delay to the higher priority customer 3 (in view of the exhaustive nature of the dispatching policy).

Figure 1. The optimal dispatching actions when the repairman's current location is $l=1$ and the repair queue length vector q is such that $q_1=0, q_2=0, q_3>0, q_4>0$.

		q_4							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
q_3	**1**	4	4	4	4	4	4	4	4
	2	4	4	4	4	4	3	3	3
	3	3	3	3	3	3	3	3	3
	4	3	3	3	3	3	3	3	3
	5	3	3	3	3	3	3	3	3

Figure 2. The optimal dispatching actions when the repairman's current location is $l=3$ and the repair queue length vector q is such that $q_1>0, q_2=0, q_3=0, q_4>0$.

		q_4							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
q_1	**1**	4	4	4	4	4	4	4	4
	2	4	4	1	1	1	1	1	1
	3	1	1	1	1	1	1	1	1
	4	1	1	1	1	1	1	1	1
	5	1	1	1	1	1	1	1	1
	6	1	1	1	1	1	1	1	1

Now, we show that a naïve dispatching policy can severely deviate from optimum. Consider the "nearest neighbor" policy which selects the nearest location with nonempty repair queue for the next visit. This policy has an expected long-run average downtime cost of \$17.03 per time unit, which is around 26% away from the minimal average cost $g(\pi^*)$.

V. Conclusions

We have considered the problem of dispatching a traveling repairman to different locations with unreliable machines. The objective has been to obtain a dynamic dispatching policy minimizing the overall downtime penalties. The repairman operations are modeled as a multiclass finite population queueing system. Then, a semi-Markov decision process model has been formulated to find the optimal policy. A numerical example is provided to illustrate our approach. In particular, the optimal dispatching is not necessarily monotone in the number of machine failures. Moreover, it is shown that a naïve dispatching policy such as the nearest neighbor can perform poorly compared to the optimal policy. As a direction for future research, it would be interesting to consider a more general machine failure time distribution such as the Erlang distribution.

Acknowledgements

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

Appendix

Here, we derive both $\tau_n(l,q,a)$ and $\rho_n(l,q,a)$. Suppose the repairman initiates a repair at his current location l. Let $f_{S_l}(\cdot)$ and $F_{S_l}(\cdot)$ be the pdf and cdf corresponding to the random repair time S_l. Let B_n be the random time-to failure of an operating customer n machine. Then,

$$\begin{aligned} \tau_n(l,q,a) &= \mathbb{E}\{\textstyle\int_0^\infty \mathbb{I}\{B_n<t, S_l>t\}dt\} \\ &= \textstyle\int_0^\infty \mathbb{E}\{\mathbb{I}\{B_n<t, S_l>t\}\}dt \\ &= \textstyle\int_0^\infty \mathbb{P}\{B_n<t, S_l>t\}dt \\ &= \textstyle\int_0^\infty \mathbb{P}\{B_n<t\}\mathbb{P}\{S_l>t\}dt \\ &= \textstyle\int_0^\infty (1-\exp\{-\lambda_n t\})\left(1 - F_{S_l}(t)\right)dt \\ &= \textstyle\int_0^\infty \left(1-F_{S_l}(t)\right)dt - \int_0^\infty \exp\{-\lambda_n t\}\left(1 - F_{S_l}(t)\right)dt \\ &= \bar{S}_l - (1-\tilde{S}_l(\lambda_n))/\lambda_n \end{aligned} \tag{13}$$

Most notably, the 4th expression of (13) holds because the time-to-failure and the repair completion times are independent, i.e. B_n and S_l are independent. The 5th expression holds because B_n is an exponential random variable with parameter λ_n. The last expression is a result of a basic property of Laplace transforms. Also,

$$\begin{aligned} \rho_n(l,q,a) &= \mathbb{P}\{B_n<S_l\} \\ &= \textstyle\int_0^\infty \mathbb{P}\{B_n<S_l \mid S_l=t\} f_{S_l}(t)dt \\ &= \textstyle\int_0^\infty (1-\exp\{-\lambda_n t\}) f_{S_l}(t)dt \\ &= \textstyle\int_0^\infty f_{S_l}(t)dt - \int_0^\infty \exp\{-\lambda_n t\} f_{S_l}(t)dt \\ &= 1-\tilde{S}_l(\lambda_n) \end{aligned} \tag{14}$$

The 2nd expression of (14) is an application of the law of total probability, while the last expression follows by definition of a Laplace transform.

With the same line of thinking, it holds that $\tau_n(l,q,a) = \bar{T}_{l,a} - (1 - \tilde{T}_{l,a}(\lambda_n))/\lambda_n$ and that $\rho_n(l,q,a) = 1 - \tilde{T}_{l,a}(\lambda_n)$ when the repairman travels from l to a.

References

[1] H. Tang, E. Miller-Hooks, and R. Tomastik, "Scheduling technicians for planned maintenance of geographically distributed equipment," Transportation Res. Part E: Logistics and Transportation Review, vol. 43, no. 5, pp. 591-609, September 2007.

[2] J.F. Cordeau, G. Laporte, F. Pasin, and S. Ropke, "Scheduling technicians and tasks in a telecommunications company," J. Sched., vol. 13, no. 4, pp. 393-409, August 2010.

[3] S, Binart, P. Dejax, M. Gendreau, and F. Semet, "A 2-stage method for a field service routing problem with stochastic travel and service times," Comp. Oper. Res., vol. 65, pp. 64-75, January 2016.

[4] E. López-Santana, R. Akhavan-Tabatabaei, L. Dieulle, N. Labadie, and A.L. Medaglia, "On the combined maintenance and routing optimization problem," Rel. Eng. Syst. Saf., vol. 145, pp. 199-214, January 2016.

[5] H.N. Psaraftis, M. Wen, and C.A. Kontovas, "Dynamic vehicle routing problems: three decades and counting," Networks, vol. 67, pp. 3-31, August 2015.

[6] D.J. Bertsimas and G. van Ryzin, "A stochastic and dynamic vehicle routing problem in the Euclidean plane," Oper. Res., vol. 39, no. 4, pp. 601-615, February 1991.

[7] M.A.A. Boon, R.D. van der Mei, and E.M.M. Winands, "Applications of polling systems," Surv. Oper. Res. Manag. Sci., vol. 16, no. 2, pp. 67-82, March 2011.

[8] H. Tijms, Stochastic Modelling and Analysis: A Computational Approach. Chichester: Wiley, 1986.