

Outsourcing Warranty Repairs: Dynamic Allocation

Michelle Opp[†]

Kevin Glazebrook^{‡*}

Vidyadhar G. Kulkarni^{†**}

† Department of Statistics and Operations Research
University of North Carolina
Chapel Hill, NC 27599

‡ The Management School
University of Edinburgh
Edinburgh EH8 9JY

July 26, 2004

Abstract

In this paper we consider the problem of minimizing the costs of outsourcing warranty repairs when failed items are dynamically routed to one of several service vendors. In our model, the manufacturer incurs a repair cost each time an item needs repair and also incurs a goodwill cost while an item is awaiting and undergoing repair. For a large manufacturer with annual warranty costs in the tens of millions of dollars, even a small relative cost reduction from the use of dynamic (rather than static) allocation may be practically significant. However, due to the size of the state space, the resulting dynamic programming problem is not exactly solvable in practice. Furthermore, standard routing heuristics, such as join-the-shortest-queue, are simply not good enough to identify potential cost savings of any significance. We use two different approaches to develop effective, simply structured index policies for the dynamic allocation problem. The first uses dynamic programming policy improvement while the second deploys Whittle's proposal for restless bandits. The closed form indices concerned are new and the policies sufficiently close to optimal to provide cost savings over static allocation. All results of this paper are demonstrated using a simulation study.

Key words: Optimal allocation, Warranty outsourcing, Index policies, Dynamic routing, Restless bandit.

*Partially supported by the Engineering and Physical Sciences Research Council through grant GR/S45188/01.

**Partially supported by NSF grant DMI-0223117.

1 Introduction

In recent years, the trend of outsourcing warranty repairs has seen enormous growth. In particular, this practice is common in the PC industry, where manufacturers contract outside vendors to repair items that fail within the warranty period. In doing so, a manufacturer can often improve turnaround times by using geographically distributed vendors, and can also decrease costs by not having to maintain an in-house repair facility.

On the other hand, outsourcing warranty repairs also increases the manufacturer's exposure to risk in terms of customer satisfaction, which may lead to future lost sales. Therefore, the manufacturer must find a balance between low costs and acceptable customer service levels while managing the outsourced warranty repair services.

In this paper, we consider the following scenario: A large manufacturer sells items with a warranty, the length of which is specified in the contract. Any needed repairs that are performed while the item is under warranty are at no charge to the customer; the manufacturer and/or the service vendor absorbs the entire cost of the repair. In order to service all the customers and to prevent long delays for customers, the manufacturer outsources to several service vendors.

Opp et al. [25] consider the problem of minimizing the costs of outsourcing warranty repairs to alternative service vendors using a static allocation model. That is, there is a fixed number of items under warranty; at the beginning of the warranty period, each item is preassigned to one of the service vendors. Then, each time an item requires repair, it is sent to its preassigned service vendor for repair. In this paper, we consider the dynamic allocation of items to vendors. In this case, whenever an item fails, the customer calls a central office, where the central decision maker uses information about the current state at each service vendor to decide which vendor will be used to repair that failure. Therefore, an item may be repaired by one vendor for its first failure under warranty, but may be repaired by a different vendor for the next failure under warranty. Because the manufacturer delays the decisions until the times of failure when more information about the congestion at each vendor is known, we expect dynamic routing to produce lower-cost policies than the static allocation given in Opp et al. [25].

For large manufacturers, annual warranty costs can amount to tens of millions of dollars. Therefore, even a small relative cost reduction from the use of dynamic (rather than static) allocation may be practically significant. However, the size of the state space means that the resulting dynamic programming problem is not exactly solvable in practice. Furthermore, standard routing heuristics, such as join-the-shortest-queue,

do not take into account the particular cost structure for this problem, and are simply not good enough to identify potential cost savings of any significance. We use two different approaches to develop effective, simply structured index policies for the dynamic allocation problem. The first uses dynamic programming (DP) policy improvement while the second deploys Whittle’s proposal for restless bandits. The indices concerned are new and the policies sufficiently close to optimal to provide cost savings over static allocation. All results of this paper are demonstrated using a simulation study.

The rest of this paper is organized as follows: In Section 2, we define the notation and describe the simple static allocation model, which can be used as a comparison with the dynamic allocation policies to be developed later in the paper. In Section 3, we formulate the routing problem as a continuous-time Markov decision process (CTMDP). We then proceed to develop index policies from the policy improvement and restless bandit approaches in Sections 4 and 5, respectively. Through a detailed simulation study in Section 6, we compare the index policies to the optimal static allocation (Section 6.2), the optimal dynamic routing policy (Section 6.3), and two simple dynamic routing heuristics (Section 6.4). Section 7 contains some concluding remarks.

2 Static Allocation Model

In this section, we describe the simple static allocation model in which each item is preassigned to one of the repair vendors. Full details about this model and the solution method can be found in Opp et al. [25]. In the static allocation model, the manufacturer first decides how many items to allocate to each repair vendor. Then, each time an item fails, it is sent to the preassigned vendor for repair. The motivation behind a static allocation model lies in its simplicity and ease of implementation, as this type of static model results in a static, deterministic routing policy. In addition, full information about the current state is not always known, in which case a dynamic allocation policy cannot be implemented. Static allocation models are common in load balancing (Combé and Boxma [4], Hordijk, Loeve, and Tiggelman [16], Cheng and Muntz [3]) and server allocation (Rolfe [28], Dyer and Proll [6], Shanthikumar and Yao [30], [31]), among other areas.

Using a static policy, the manufacturer outsources warranty repairs for K identical items to V service vendors. Vendor i ($i = 1, \dots, V$) has s_i identical servers, each with exponential service times with rate μ_i . The time between failures for a single item is exponentially distributed with rate λ . We assume that the information about λ , μ_i , and s_i is known to the manufacturer.

For each repair performed by vendor i , the manufacturer must pay the vendor a fixed amount c_i . The manufacturer must also consider the loss of customer goodwill associated with long waits for repair. To account for this, the manufacturer incurs a goodwill cost at a rate of h_i per unit time that an item spends in queue and service at vendor i .

The decision variables in the static allocation model are k_i , the number of items to allocate to vendor i in order to minimize expected total warranty cost. For a fixed value k_i , the repair process at vendor i is modelled as an $M/M/s_i/\infty/k_i$ queue with arrival rate λ , service rate μ_i , and finite arrival population k_i .

To express the total cost to the manufacturer, first let $L_i(k_i)$ be the expected number of items (customers) at vendor i when the allocation to vendor i is k_i . Computing $L_i(k_i)$ directly from the probability distribution is tedious and time-consuming; however, one can recursively compute $L_i(k_i)$ using mean value analysis as described in Opp et al. [25].

The manufacturer must pay a fixed cost c_i to vendor i each time an item is sent to vendor i for repair, and also incurs the goodwill cost h_i per unit time that the item remains at vendor i . Therefore, the manufacturer's expected cost per unit time for repairs at the i th vendor, denoted by $f_i(k_i)$, is given as follows:

$$\begin{aligned} f_i(k_i) &= c_i\lambda(k_i - L_i(k_i)) + h_iL_i(k_i) \\ &= \lambda c_i k_i + (h_i - \lambda c_i)L_i(k_i). \end{aligned}$$

The resulting optimization problem is a resource allocation problem with integer variables (see Gross [14], Fox [9], Ibaraki and Katoh [17]):

$$\begin{aligned} &\text{Minimize} && \sum_{i=1}^V f_i(k_i) \\ &\text{subject to} && \sum_{i=1}^V k_i = K, \\ &&& k_i \geq 0 \text{ and integer, } \quad i = 1, \dots, V. \end{aligned}$$

The convexity of the objective function term $f_i(k_i)$ is established in Opp et al. [25], using the concavity of throughput from Dowdy et al. [5]. Hence, it follows that $f_i(k_i)$ is convex if $h_i \geq \lambda c_i$. When this is true for all $i = 1, \dots, V$, the static allocation problem is a separable convex resource allocation problem, and the optimal allocation can be found using a greedy algorithm, first proposed by Gross [14]; see also Fox [9].

Greedy Algorithm for Optimal Static Allocation

- **Step 0:** Set $k_i = 0$ for $i = 1, \dots, V$.
- **Step 1:** Choose a $j \in \operatorname{argmin}_{i=1, \dots, V} \{f_i(k_i + 1) - f_i(k_i)\}$.
- **Step 2:** Set $k_j = k_j + 1$.
- **Step 3:** If $\sum_{i=1}^V k_i < K$, go to Step 1. Else, stop: (k_1, \dots, k_V) is optimal.

Therefore, the optimal static allocation for the convex case is quite simple to compute. However, the static allocation model ignores important information about the current level of congestion at each vendor, which contributes to the goodwill (holding) cost. We now turn to the related dynamic allocation model for the warranty outsourcing problem, and we use a simulation model to compare the optimal static allocation with the dynamic index policies derived in the following sections.

3 Dynamic Model Formulation

We model the dynamic warranty outsourcing problem as a routing control problem in a closed queueing network with finite population K , as depicted in Figure 1. Station 0 includes all items that are properly functioning (that is, not undergoing or awaiting repair); this station can be thought of as a multi-server queue with K servers, each server having exponential service times with rate λ (the failure rate of the items). When an item fails, the central decision-maker (denoted by D in Figure 1) decides, based on the costs and congestions at each vendor, to which vendor the item is sent. Station i represents the i th service vendor ($i = 1, \dots, V$); this station has s_i servers, each with exponential service times with rate μ_i . In addition, an item sent to station i incurs a fixed cost c_i , as well as a per-unit-time holding cost h_i while it remains at station i .

The routing control problem for two parallel single-server queues with infinite population has been studied in great detail. Under certain assumptions regarding the cost and service rate parameters, the optimal routing decisions in this case have been shown to satisfy routing monotonicity, resulting in a routing policy of threshold type. Ephremides, Varaiya, and Walrand [7] consider the case of two similar queues; that is, the service rates of the two queues are equal, and both queues incur the same holding cost and zero fixed cost. They show that if the queue lengths are observable, then the “join the shortest queue” (JSQ) rule is optimal. Furthermore, this result extends to more than two queues, as long as the service rates and costs

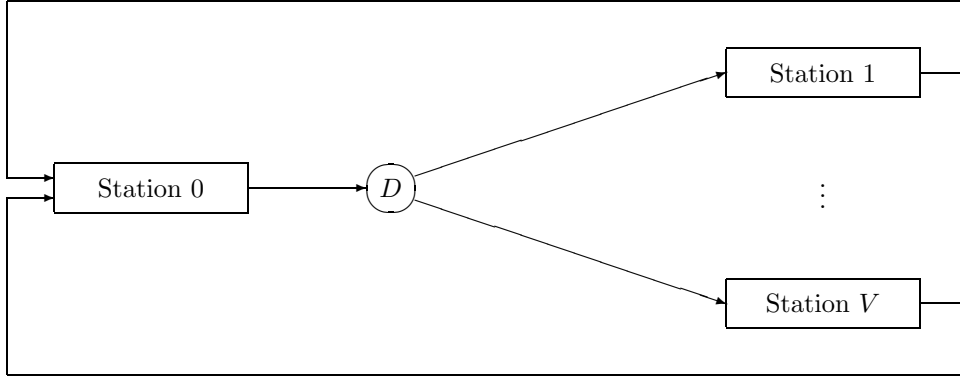


Figure 1: Dynamic routing with closed population

are identical at all queues. Hajek [15] extends this result with an inductive proof of routing monotonicity when the service rates are not equal.

Stidham and Weber [32] provide a survey of results regarding control of networks of queues using Markov decision models. They discuss not only routing control, but also admission control, service rate control, and server allocation, among other topics. Combining the admission and routing control models into one framework, routing monotonicity holds if the fixed costs for routing to each queue are equal (this corresponds to a constant cost of admitting a customer to the system) and the holding costs at each queue are equal, regardless of whether the service rates are equal.

When the number of items to be covered under warranty is large and the failure rate is comparatively low, the finite-source dynamic routing problem with two vendors can be approximated by the infinite-source routing control problem, and we would therefore expect similar switching curve results. To our knowledge, however, there has been no work done for the general model with more than two multi-server queues and a closed population for arrivals under this particular cost structure.

We define the CTMDP as follows. Let $X_i(t)$ denote the number of items undergoing or awaiting repair at vendor i at time t ($i = 1, \dots, V$; $t \geq 0$). We say that vendor i is in state x_i at time t if $X_i(t) = x_i$; the state of the system is denoted by $\mathbf{x} = (x_1, \dots, x_V)$. Because we are considering a closed population, the state space of $X(t) = [X_1(t), \dots, X_V(t)]$ is given by $S = \{\mathbf{x} = (x_1, \dots, x_V) \in \mathbb{Z}^V : x_i \geq 0, \sum_{i=1}^V x_i \leq K\}$. The action space is given by $A = \{1, \dots, V\}$, where action $i \in A$ indicates that an incoming failed item is sent to vendor i for repair.

To simplify the notation, let $\mu_i(x_i) = \mu_i \min(x_i, s_i)$, and let \mathbf{e}_i denote the i th unit vector (that is, \mathbf{e}_i is the i th row of the $V \times V$ identity matrix). In state \mathbf{x} , new failures occur at rate $\lambda \left(K - \sum_{i=1}^V x_i \right)$, and repair

completions occur at rate $\sum_{i=1}^V \mu_i(x_i)$, for a total transition rate given by

$$\lambda \left(K - \sum_{i=1}^V x_i \right) + \sum_{i=1}^V \mu_i(x_i).$$

When an incoming failure is routed to vendor i , the manufacturer incurs a fixed cost c_i , and the state changes from \mathbf{x} to $\mathbf{x} + \mathbf{e}_i$. When a repair completion occurs at vendor i , the state changes from \mathbf{x} to $\mathbf{x} - \mathbf{e}_i$. The holding cost rate in state \mathbf{x} is given by $\sum_{i=1}^V h_i x_i$.

Following the standard course of uniformization, we choose a suitable time scale so that $K\lambda + \sum_{i=1}^V \mu_i s_i = 1$. We introduce “fictitious” transitions in state \mathbf{x} (which result in no change of state) so that the total transition rate out of state \mathbf{x} is 1 (Lippman [21]). A fictitious transition in state \mathbf{x} occurs at the following rate:

$$\begin{aligned} & 1 - \lambda \left(K - \sum_{i=1}^V x_i \right) - \sum_{i=1}^V \mu_i(x_i) \\ &= K\lambda + \sum_{i=1}^V \mu_i s_i - \lambda \left(K - \sum_{i=1}^V x_i \right) - \sum_{i=1}^V \mu_i(x_i) \\ &= \sum_{i=1}^V (\lambda x_i + \mu_i s_i - \mu_i(x_i)). \end{aligned}$$

Let $g_\pi(\mathbf{x})$ denote the long-run average cost associated with state \mathbf{x} under policy π and let $w_\pi(\mathbf{x})$ denote the bias associated with state \mathbf{x} under policy π . Because the state space S and the action space A are both finite, we have the following theorem, which is based on Proposition 2.1 in Bertsekas [2].

Theorem 1. *If a scalar g and a vector \mathbf{w} satisfy*

$$\begin{aligned} g + w(\mathbf{x}) &= \sum_{i=1}^V h_i x_i + \sum_{i=1}^V \mu_i(x_i) w(\mathbf{x} - \mathbf{e}_i) \\ &+ \sum_{i=1}^V (\lambda x_i + \mu_i s_i - \mu_i(x_i)) w(\mathbf{x}) \\ &+ \lambda \left(K - \sum_{i=1}^V x_i \right) \min_{j=1, \dots, V} \{c_j + w(\mathbf{x} + \mathbf{e}_j)\} \end{aligned} \tag{1}$$

for all $\mathbf{x} \in S$, then g is the optimal average cost per stage for all \mathbf{x} . Furthermore, if $\pi^*(\mathbf{x})$ attains the minimum in Eq. (1) for each \mathbf{x} , the stationary policy π^* is optimal.

For small instances of the problem (e.g., $K \leq 300$ and $V = 2$), we can solve the optimality equations to obtain the optimal long-run average cost g (which we do in Section 6.3). However, for larger values of K or V , finding an exact solution to the DP equations is usually numerically intractable. We therefore find nearly optimal policies using two different index policies: one derived from policy improvement (described in Section 4), and the other derived from restless bandit models (described in Section 5).

4 Policy-Improvement Approach

In this section, we use a policy-improvement approach to develop an approximately optimal routing policy for this problem. The derived heuristic will assign to each vendor a function of its current state (called the *index*), and will route a new failure to the vendor with the smallest index.

First, we assume that $\lambda \rightarrow 0$ and $K \rightarrow \infty$ such that $K\lambda \rightarrow \bar{\lambda}$, a constant. In practice, the population size is sufficiently large and the failure rate is sufficiently small that the dynamically changing arrival rate $\lambda \left(K - \sum_{i=1}^V x_i \right)$ can be approximated by a constant arrival rate $\bar{\lambda}$ between decision epochs. We develop the index policy using this fixed arrival rate $\bar{\lambda}$; it is easy to perform a post hoc adjustment for the actual varying arrival rate in the calculations of the index policy values.

With this assumption, the average-cost optimality equations in (1) can be modified to the following (with uniformization $\bar{\lambda} + \sum_{i=1}^V \mu_i s_i = 1$):

$$\begin{aligned}
 g + w(\mathbf{x}) &= \sum_{i=1}^V h_i x_i + \sum_{i=1}^V \mu_i(x_i) w(\mathbf{x} - \mathbf{e}_i) \\
 &\quad + \sum_{i=1}^V (\mu_i s_i - \mu_i(x_i)) w(\mathbf{x}) \\
 &\quad + \bar{\lambda} \min_{j=1, \dots, V} \{c_j + w(\mathbf{x} + \mathbf{e}_j)\}.
 \end{aligned} \tag{2}$$

One method of solving (2) is via the policy improvement algorithm. However, even with a fixed arrival rate, performing several iterations of policy improvement is usually numerically intractable for problems of realistic size. We follow Krishnan [18] in proposing the development of dynamic routing heuristics by the application of a single policy improvement step applied to an optimal state-independent policy. See also the discussion of Tijms [33]. One of the major contributions of the paper is the demonstration that this results in a simple index heuristic for routing, which we develop in simple closed form. Hence, each vendor i has an associated calibrating index I_i , a function of the number of repairs x_i currently waiting at vendor i . At each arrival epoch, the heuristic sends the new item for repair to the vendor with smallest index.

4.1 Choosing an Initial Policy

The first step in policy improvement is to choose an initial policy for the problem; we choose an optimal state-independent policy as the initial policy. A state-independent policy $\mathbf{p} = (p_1, \dots, p_V)$ routes an incoming failure to vendor i with probability p_i , independent of the state of the system. Under this policy, vendor i sees an incoming Poisson stream of customers with rate $\bar{\lambda}p_i$; therefore, vendor i can be viewed as an $M/M/s_i$ system. Note that we are assuming $\sum_{i=1}^V s_i\mu_i > \bar{\lambda}$; that is, the total workload of all vendors is enough to handle the incoming customer stream. As a consequence, we know that there exist policies \mathbf{p} such that $\bar{\lambda}p_i < s_i\mu_i$ for all $i = 1, \dots, V$. In what follows, we only consider such stable policies. The expected long-run average cost of policy \mathbf{p} is given by

$$\sum_{i=1}^V (c_i\bar{\lambda}p_i + h_iL_i(\bar{\lambda}p_i)), \quad (3)$$

where $L_i(\bar{\lambda}p_i)$ is the expected number of customers in steady state in an $M/M/s_i$ system with arrival rate $\bar{\lambda}p_i$. From Kulkarni [19], this is

$$L_i(\bar{\lambda}p_i) = \frac{\bar{\lambda}p_i}{\mu_i} + \frac{\alpha_{i,p}\rho_{i,p}}{(1 - \rho_{i,p})^2}, \quad (4)$$

where $\alpha_{i,p}$ is the steady-state probability of exactly s_i customers in an $M/M/s_i$ system with arrival rate $\bar{\lambda}p_i$, and $\rho_{i,p} = \bar{\lambda}p_i/s_i\mu_i$. For $s_i = 1$, this simplifies to $L_i(\bar{\lambda}p_i) = \bar{\lambda}p_i/(\mu_i - \bar{\lambda}p_i)$.

$L_i(\bar{\lambda}p_i)$ is a convex function of $\bar{\lambda}p_i$ (Grassmann [13], Lee and Cohen [20]), and hence a convex function of p_i . Therefore, $c_i\bar{\lambda}p_i + h_iL_i(\bar{\lambda}p_i)$ is also convex, and the problem of minimizing the objective (3) subject to the constraint $\sum_{i=1}^V p_i = 1$ is a separable convex resource allocation problem with continuous variables. To find the optimal solution, denoted \mathbf{p}^* , we use the ranking algorithm described in Ibaraki and Katoh [17]. This algorithm was first proposed by Luss and Gupta [23]; the algorithm presented in Ibaraki and Katoh [17] is a refined version due to Zipkin [37]. We then use the state-independent policy \mathbf{p}^* as the initial policy in the policy improvement algorithm.

4.2 Policy Improvement Step

Let \hat{g} and $\hat{w}(\mathbf{x})$ denote the long-run average cost and bias, respectively, of the state-independent policy \mathbf{p}^* . These are given by the solution to the following system of equations:

$$\begin{aligned}
\hat{g} + \hat{w}(\mathbf{x}) &= \sum_{i=1}^V h_i x_i + \sum_{i=1}^V \mu_i(x_i) \hat{w}(\mathbf{x} - \mathbf{e}_i) \\
&\quad + \sum_{i=1}^V (\mu_i s_i - \mu_i(x_i)) \hat{w}(\mathbf{x}) \\
&\quad + \bar{\lambda} \sum_{i=1}^V p_i^* (c_i + \hat{w}(\mathbf{x} + \mathbf{e}_i)).
\end{aligned}$$

We improve this policy by implementing a single dynamic programming (DP) policy improvement step. The improved policy is the one that, in state \mathbf{x} , chooses a vendor j that minimizes $c_j + \hat{w}(\mathbf{x} + \mathbf{e}_j)$. This is equivalent to choosing a vendor j that minimizes $c_j + \hat{w}(\mathbf{x} + \mathbf{e}_j) - \hat{w}(\mathbf{x})$.

We have from the theory of Markov decision processes (MDPs) that

$$\hat{w}(\mathbf{x} + \mathbf{e}_j) - \hat{w}(\mathbf{x}) = K_j(x_j + 1) - K_j(x_j) - g_j^*(\bar{\lambda} p_j^*)(T_j(x_j + 1) - T_j(x_j)), \quad (5)$$

where the notation $K_i(x_i)$ is used for the expected cost incurred at vendor i from an initial state x_i until the vendor reaches state 0 for the first time, $T_i(x_i)$ is the corresponding expected time, and $g_i^*(\bar{\lambda} p_i^*)$ is the average cost per unit time at vendor i under fixed arrival rate $\bar{\lambda} p_i^*$. We have that

$$g_i^*(\bar{\lambda} p_i^*) = c_i \bar{\lambda} p_i^* + h_i L_i(\bar{\lambda} p_i^*).$$

The calculation which yields (5) makes extensive use of the fact that entry into state 0 is a regeneration point for the process concerned (Kulkarni [19]).

We now define

$$I_j(x_j) = c_j + K_j(x_j + 1) - K_j(x_j) - g_j^*(\bar{\lambda} p_j^*)(T_j(x_j + 1) - T_j(x_j)). \quad (6)$$

A closed form solution for $I_j(x_j)$ is given in the following theorem, using $\gamma_j = \bar{\lambda} p_j^*$, $\rho_j = \gamma_j / (s_j \mu_j)$, and

$$\alpha_j = \frac{\frac{1}{s_j!} \left(\frac{\gamma_j}{\mu_j} \right)^{s_j}}{\sum_{n=0}^{s_j-1} \frac{1}{n!} \left(\frac{\gamma_j}{\mu_j} \right)^n + \frac{s_j^{s_j}}{s_j!} \left(\frac{\rho_j^{s_j}}{1 - \rho_j} \right)}. \quad (7)$$

Theorem 2 (Index Policy for Dynamic Routing: Policy Improvement). *The dynamic policy obtained upon implementing a single policy improvement step from the optimal static policy \mathbf{p}^* operates as follows: In state \mathbf{x} , route an incoming repair to any vendor i such that*

$$I_i(x_i) = \min_{1 \leq j \leq V} I_j(x_j),$$

where

$$I_j(x_j) = \begin{cases} c_j + \frac{h_j}{\mu_j} + x_j! \left(\frac{\mu_j}{\gamma_j} \right)^{x_j} \frac{h_j \alpha_j \rho_j}{\gamma_j (1 - \rho_j)^2} \sum_{n=0}^{x_j} \frac{(\gamma_j / \mu_j)^n}{n!}, & 0 \leq x_j \leq s_j - 1, \\ c_j + \frac{h_j}{s_j \mu_j - \gamma_j} \left(x_j + 1 + \frac{\gamma_j}{s_j \mu_j - \gamma_j} - \frac{\gamma_j}{\mu_j} - \frac{\alpha_j \rho_j}{(1 - \rho_j)^2} \right), & x_j \geq s_j. \end{cases} \quad (8)$$

Proof. For notational convenience, we drop the vendor suffix j and write γ , ρ , α , and $g^*(\gamma)$ in place of γ_j , ρ_j , α_j , and $g_j^*(\gamma_j)$, respectively. It is clear that α of equation (7) is the probability that there are exactly s customers in an $M/M/s$ queue with arrival rate γ and service rate μ .

Let $L(\gamma)$ be the expected number of customers in this system, as given in equation (4). Then

$$\begin{aligned} g^*(\gamma) &= c\gamma + hL(\gamma) \\ &= c\gamma + \frac{h\gamma}{\mu} + \frac{h\alpha\rho}{(1 - \rho)^2}. \end{aligned} \quad (9)$$

Now let $\mu_x = \mu \min(x, s)$. Using first-step analysis, the expected time $T(x)$ is given by the solution to the following difference equations:

$$T(x) = \frac{1}{\gamma + \mu_x} + \frac{\mu_x}{\gamma + \mu_x} T(x - 1) + \frac{\gamma}{\gamma + \mu_x} T(x + 1), \quad (10)$$

with $T(0) = 0$. Similarly, the expected cost $K(x)$ is given by the solution to

$$K(x) = \frac{hx}{\gamma + \mu_x} + \frac{\mu_x}{\gamma + \mu_x} K(x - 1) + \frac{\gamma}{\gamma + \mu_x} (c + K(x + 1)), \quad (11)$$

with $K(0) = 0$. We use equations (10) and (11) to derive the closed form solution for $I(x)$ by considering two cases.

Case 1: $0 \leq x \leq s - 1$

For $0 \leq x \leq s$, equations (10) and (11) give

$$\mu_x \{T(x) - T(x - 1)\} = 1 + \gamma \{T(x + 1) - T(x)\}, \quad (12)$$

$$\mu_x \{K(x) - K(x - 1)\} = (hx + \gamma c) + \gamma \{K(x + 1) - K(x)\}. \quad (13)$$

Let $\psi(x) = I(x) - c - h/\mu$. Using (9), (12), and (13) in (6) and simplifying, we get

$$\psi(x) = \frac{\mu x}{\gamma} \psi(x-1) + \frac{h\alpha\rho}{\gamma(1-\rho)^2}. \quad (14)$$

Next we evaluate $\psi(0)$. Observe that the expected cost incurred by the process under study during each busy period initiated by a single repair in the system is $c + K(1) = c + K(1) - K(0)$, while the expected time between the starts of successive busy periods after the first is $1/\gamma + T(1) = 1/\gamma + T(1) - T(0)$. It follows from a standard renewal theory argument that

$$g^*(\gamma) = \frac{c + K(1) - K(0)}{1/\gamma + T(1) - T(0)},$$

where $K(0) = T(0) = 0$. Therefore

$$\begin{aligned} I(0) &= c + K(1) - K(0) - g^*(\gamma) \{T(1) - T(0)\} \\ &= \frac{g^*(\gamma)}{\gamma} \\ &= c + \frac{h}{\mu} + \frac{h\alpha\rho}{\gamma(1-\rho)^2}. \end{aligned}$$

Hence

$$\psi(0) = \frac{h\alpha\rho}{\gamma(1-\rho)^2}.$$

Solving (14) recursively, we get

$$\psi(x) = x! \left(\frac{\mu}{\gamma}\right)^x \frac{h\alpha\rho}{\gamma(1-\rho)^2} \sum_{i=0}^x \frac{(\gamma/\mu)^i}{i!}$$

for $0 \leq x \leq s-1$. The result follows.

Case 2: $x \geq s$

For $x \geq s$, we have $\mu_x = s\mu$, and equation (11) reduces to

$$s\mu \{K(x) - K(x-1)\} = hx + \gamma \{c + K(x+1) - K(x)\}.$$

Solving the above difference equation, we get

$$K(x+1) - K(x) = \frac{h(x+1)}{s\mu - \gamma} + \frac{c\gamma}{s\mu - \gamma} + \frac{h\gamma}{(s\mu - \gamma)^2}.$$

Similarly, solving (10) with $\mu_x = s\mu$ gives

$$T(x+1) - T(x) = 1/(s\mu - \gamma).$$

Substituting the expressions for $K(x+1) - K(x)$, $T(x+1) - T(x)$ and $g^*(\gamma)$ in equation (6) yields the result from Theorem 2 for $x \geq s$. \square

Note that the index $I_j(x_j)$ is increasing linear in the workload for the range of importance $x_j \geq s_j$. In addition, for the special case $s_j = 1$, the index can be simplified to the following:

$$I_j(x_j) = c_j + \frac{h_j(x_j + 1)}{\mu_j - \gamma_j}.$$

In practice, we use $\lambda \left(K - \sum_{i=1}^V x_i \right)$ in place of $\bar{\lambda}$ to compute the policy with indices $I_j(x_j)$. That is, we modify the policy to account for the dynamically changing arrival rate $\lambda \left(K - \sum_{i=1}^V x_i \right)$, rather than assuming a fixed arrival rate $\bar{\lambda}$. In this case, the arrival rate γ_j in (8) is calculated as $\gamma_j = \lambda p_j^* \left(K - \sum_{i=1}^V x_i \right)$. Note, however, that the original definition of p_j^* does not change; that is, p_j^* is calculated at the beginning, assuming a fixed arrival rate $\bar{\lambda}$. This value is subsequently used in the calculation of the index for vendor j when accounting for the dynamic arrival rate $\lambda \left(K - \sum_{i=1}^V x_i \right)$. This modification is very easy to implement in the index calculation, and results in a nearly optimal policy, as demonstrated in Section 6.

5 Restless Bandit Approach

Whittle [35] introduced a class of models for stochastic resource allocation called *restless bandits*. These are generalizations of the classic multi-armed bandits of Gittins [10] which allow evolution of projects even when not in receipt of service. This class of processes has been shown to be PSPACE-hard by Papadimitriou and Tsitsiklis [26], which almost certainly rules out optimal policies of simple form. Whittle himself described an approach to the development of *index heuristics* for restless bandits which centered around Lagrangian relaxations of the original problem. Subsequent studies have elucidated, both theoretically and empirically, the strong performance of Whittle's index policy. See, for example, Ansell et al. [1], Glazebrook, Niño-Mora, and Ansell [12], and Weber and Weiss [34]. Whittle [36] proposed the deployment of restless bandit approaches to the development of dynamic policies for the routing of customers to alternative service stations. Niño-Mora [24] has developed a general theory which extends Whittle's ideas and discusses when they can be successfully applied to routing problems. Infinite population approximations to our models satisfy all of the sufficient conditions concerned and hence we can in principle develop index heuristics for the problems discussed here using Whittle's ideas. We now proceed to describe the main ideas underlying this approach and will then proceed to develop the indices concerned in closed form.

Whittle's indices are properties of *individual vendors* and hence we focus the following discussion on one such vendor, labeled j . To develop the index, we suppose that vendor j is facing the entire incoming stream of repairs, which has rate $\bar{\lambda} = K\lambda$ since we will again consider the infinite-population problem while developing the index. The vendor has the freedom to accept or reject each incoming customer. These two actions correspond respectively to routing the incoming repair to vendor j (accept) or to another vendor (reject) in the full multi-vendor problem. The economic structure of this single-vendor problem includes the repair costs (c_j) and holding costs (with rate h_j) discussed in Section 3, but these are enhanced by a rejection penalty W which is payable whenever an incoming customer is rejected for service. Write $\pi_j(W)$ for a general stationary policy for accepting/rejecting incoming customers. The single-vendor problem with rejection penalty W seeks $\pi_j(W)$ to minimize

$$E_{\pi_j(W)} \left[h_j X_j(t) + c_j \mathbb{I}_j \{X_j(t)\} + W (1 - \mathbb{I}_j \{X_j(t)\}) \right],$$

where

$$\mathbb{I}_j \{x_j\} = \begin{cases} 1, & \text{if a customer is accepted for service when the queue length is } x_j, \\ 0, & \text{otherwise.} \end{cases}$$

The general theory (see Niño-Mora [24]) asserts the existence of an increasing function $W_j : \mathbb{N} \rightarrow \mathbb{R}$ with the following property: For each queue length x_j it is optimal to accept an incoming customer at queue length x_j when $W \geq W_j(x_j)$ and to reject an incoming customer at queue length x_j when $W \leq W_j(x_j)$. Hence $W_j(x_j)$ may be thought of as a *fair charge* for rejection of a customer in state x_j . Whittle's index heuristic for the original multi-vendor problem always routes incoming repairs to whichever vendor has the lowest fair charge for rejection. We now describe a simple approach to the development of the indices concerned.

In order to compute $W_j(x_j)$, note that when the rejection penalty W is fixed such that $W = W_j(x_j)$, both actions of rejecting and accepting an incoming customer to vendor j are optimal in state x_j . In addition, for this W it is optimal to accept an incoming customer to vendor j for states $y_j \leq x_j - 1$, since $W_j(y_j) \leq W_j(x_j) = W$. Similarly, it is optimal to reject an incoming customer to vendor j for states $y_j \geq x_j + 1$, since $W_j(y_j) \geq W_j(x_j) = W$. It follows that $W_j(x_j)$ may be characterized as the value of the rejection penalty W that makes both of the following policies optimal for vendor j :

1. **Policy** $\pi_j(x_j)$: Accept an incoming customer to vendor j in states $\{0, 1, \dots, x_j - 1\}$, and reject an incoming customer to vendor j in states $\{x_j, x_j + 1, \dots\}$.
2. **Policy** $\pi_j(x_j + 1)$: Accept an incoming customer to vendor j in states $\{0, 1, \dots, x_j\}$, and reject an incoming customer to vendor j in states $\{x_j + 1, x_j + 2, \dots\}$.

First consider policy $\pi_j(x_j)$. Under this policy, the number of items at vendor j forms a birth-death process on the states $\{0, 1, \dots, x_j\}$, where the birth rate is given by $\bar{\lambda} = K\lambda$ for states $k = 0, \dots, x_j - 1$. The birth rate for state x_j is 0. The death rate is given by $\mu_j \min(k, s_j)$ for states $k = 1, \dots, x_j$, and the death rate for state 0 is 0.

Let $p_j(x_j, k)$ denote the steady-state probability that there are k items at vendor j under policy $\pi_j(x_j)$. For $x_j \leq s_j$, this is given by the following:

$$p_j(x_j, k) = \begin{cases} \frac{(\bar{\lambda}/\mu_j)^k / k!}{\sum_{n=0}^{x_j} (\bar{\lambda}/\mu_j)^n / n!}, & k = 0, \dots, x_j, \\ 0, & k \geq x_j + 1. \end{cases}$$

For $x_j \geq s_j + 1$, we have the following for $p_j(x_j, k)$:

$$p_j(x_j, k) = \begin{cases} \frac{(\bar{\lambda}/\mu_j)^k / k!}{\sum_{n=0}^{s_j} (\bar{\lambda}/\mu_j)^n / n! + \sum_{n=s_j+1}^{x_j} (\bar{\lambda}/\mu_j)^n / (s_j! s_j^{n-s_j})}, & k = 0, \dots, s_j, \\ \frac{(\bar{\lambda}/\mu_j)^k / (s_j! s_j^{k-s_j})}{\sum_{n=0}^{s_j} (\bar{\lambda}/\mu_j)^n / n! + \sum_{n=s_j+1}^{x_j} (\bar{\lambda}/\mu_j)^n / (s_j! s_j^{n-s_j})}, & k = s_j + 1, \dots, x_j, \\ 0, & k \geq x_j + 1. \end{cases}$$

When the vendor is in a state in which policy $\pi_j(x_j)$ accepts an incoming customer (that is, states $0, \dots, x_j - 1$), the manufacturer incurs a cost at rate $c_j \bar{\lambda}$. When the vendor is in a state in which policy $\pi_j(x_j)$ rejects an incoming customer (that is, state x_j), the manufacturer incurs a cost at rate $W \bar{\lambda}$. In all states $k = 0, \dots, x_j$, the manufacturer incurs a holding cost at rate kh_j . Therefore, the cost associated with policy $\pi_j(x_j)$ is given by

$$\begin{aligned} C_{\pi_j(x_j)}(W) &= \sum_{k=0}^{x_j-1} p_j(x_j, k)(c_j \bar{\lambda} + kh_j) + p_j(x_j, x_j)(x_j h_j + W \bar{\lambda}) \\ &= c_j \bar{\lambda} + \sum_{k=0}^{x_j} kh_j p_j(x_j, k) + \bar{\lambda}(W - c_j) p_j(x_j, x_j). \end{aligned} \quad (15)$$

Similarly, the cost associated with policy $\pi_j(x_j + 1)$ is given by

$$\begin{aligned} C_{\pi_j(x_j+1)}(W) &= \sum_{k=0}^{x_j} p_j(x_j + 1, k)(c_j \bar{\lambda} + kh_j) + p_j(x_j + 1, x_j + 1)((x_j + 1)h_j + W\bar{\lambda}) \\ &= c_j \bar{\lambda} + \sum_{k=0}^{x_j+1} kh_j p_j(x_j + 1, k) + \bar{\lambda}(W - c_j)p_j(x_j + 1, x_j + 1). \end{aligned} \quad (16)$$

The fair charge $W_j(x_j)$ is the value for which $C_{\pi_j(x_j)}(W) = C_{\pi_j(x_j+1)}(W)$; the solution to this is given in the following theorem, using

$$A_j(k) = \sum_{n=0}^k \frac{(\bar{\lambda}/\mu_j)^n}{n!}, \quad k = 0, 1, \dots$$

and

$$B_j(k) = \frac{(\bar{\lambda}/\mu_j)^k}{k!} = A_j(k) - A_j(k-1), \quad k = 0, 1, \dots$$

Theorem 3 (Index Policy for Dynamic Routing: Restless Bandit). *If $\bar{\lambda} \neq s_j \mu_j$, then $W_j(x_j)$, the fair charge for rejection in state x_j , is given by*

$$W_j(x_j) = \begin{cases} c_j + \frac{h_j}{\mu_j}, & 0 \leq x_j \leq s_j - 1, \\ c_j + h_j \left[B_j(s_j) \left(\frac{\bar{\lambda}}{\bar{\lambda} - s_j \mu_j} \right)^2 \left\{ \left(\frac{\bar{\lambda}}{s_j \mu_j} \right)^{x_j - s_j} - \left(\frac{\bar{\lambda}}{s_j \mu_j} \right)^{-1} \right\} \right. \\ \quad \left. - \left\{ B_j(s_j) \left(\frac{\bar{\lambda}}{\bar{\lambda} - s_j \mu_j} \right) - A_j(s_j) \right\} \left(x_j + 1 - \frac{\bar{\lambda}}{\mu_j} \right) \right] / \\ \quad \left[B_j(s_j) \bar{\lambda} - A_j(s_j) (\bar{\lambda} - s_j \mu_j) \right], & x_j \geq s_j. \end{cases}$$

Moreover, asymptotically as $x_j \rightarrow \infty$,

$$W_j(x_j) \sim \frac{h_j B_j(s_j) \left(\frac{\bar{\lambda}}{\bar{\lambda} - s_j \mu_j} \right)^2 \left(\frac{\bar{\lambda}}{s_j \mu_j} \right)^{x_j - s_j}}{B_j(s_j) \bar{\lambda} - A_j(s_j) (\bar{\lambda} - s_j \mu_j)} \quad \text{when } \bar{\lambda} > s_j \mu_j,$$

and

$$W_j(x_j) \sim \frac{h_j \left(x_j + 1 - \frac{\bar{\lambda}}{\mu_j} \right)}{s_j \mu_j - \bar{\lambda}} \quad \text{when } \bar{\lambda} < s_j \mu_j.$$

If $\bar{\lambda} = s_j \mu_j$, then $W_j(x_j)$ is given by

$$W_j(x_j) = \begin{cases} c_j + \frac{h_j}{\mu_j}, & 0 \leq x_j \leq s_j - 1, \\ c_j + \frac{h_j}{s_j \mu_j} \left\{ s_j + \frac{1}{2}(x_j - s_j) + \frac{1}{2}(x_j - s_j)^2 \right\}, & x_j \geq s_j. \end{cases}$$

Proof. For notational convenience, we drop the vendor subscript j , and we consider the case $\bar{\lambda} \neq s\mu$. Let

$$B_k = \frac{(\bar{\lambda}/\mu)}{k}, \quad k = 1, 2, \dots$$

Case 1: $0 \leq x \leq s - 1$

For $x \leq s - 1$, the equilibrium distribution for policy $\pi(x)$ is given by

$$p(x, k) = B(k)p(x, 0), \quad 0 \leq k \leq x,$$

where $p(x, 0)^{-1} = A(x)$. The equilibrium distribution for policy $\pi(x + 1)$ is given by

$$p(x + 1, k) = B(k)p(x + 1, 0), \quad 0 \leq k \leq x + 1,$$

where $p(x + 1, 0)^{-1} = A(x + 1)$.

The defining equation of the index $W = W(x)$ is $C_{\pi(x)}(W) = C_{\pi(x+1)}(W)$. By equating (15) and (16), this gives

$$(W - c)\bar{\lambda} \{p(x, x) - p(x + 1, x + 1)\} = \sum_{n=0}^{x+1} hnp(x + 1, n) - \sum_{n=0}^x hnp(x, n). \quad (17)$$

Multiplying both sides of (17) by $p(x, 0)^{-1}p(x + 1, 0)^{-1}$ gives

$$\begin{aligned} (W - c)\bar{\lambda} \{B(x)A(x + 1) - B(x + 1)A(x)\} = \\ h(x + 1)B(x + 1)A(x) + \sum_{n=0}^x hnB(n) \{A(x) - A(x + 1)\}. \end{aligned} \quad (18)$$

We first analyze the left side of (18), as follows:

$$\begin{aligned} (W - c)\bar{\lambda} \{B(x)A(x + 1) - B(x + 1)A(x)\} \\ = (W - c)\bar{\lambda}B(x) \{A(x + 1) - B_{x+1}A(x)\} \\ = (W - c)\bar{\lambda}B(x) \left\{ \left(1 + \sum_{y=1}^x \prod_{k=1}^y B_k \right) - B_{x+1} \left(1 + \sum_{y=1}^{x-1} \prod_{k=1}^y B_k \right) \right\} \\ = (W - c)\bar{\lambda}B(x) \left\{ 1 + \sum_{y=0}^{x-1} (B_{y+1} - B_{x+1}) \prod_{k=1}^y B_k \right\}. \end{aligned} \quad (19)$$

But

$$\begin{aligned} B_{y+1} - B_{x+1} &= \left(\frac{\bar{\lambda}}{\mu} \right) \left\{ \frac{1}{y+1} - \frac{1}{x+1} \right\} = \left(\frac{\bar{\lambda}}{(x+1)\mu} \right) \left\{ \frac{x+1}{y+1} - 1 \right\} \\ &= B_{x+1} \left\{ (x+1) \left(\frac{\mu}{\bar{\lambda}} \right) B_{y+1} - 1 \right\}. \end{aligned}$$

Therefore, (19) is equal to

$$\begin{aligned}
& (W - c)\bar{\lambda}B(x) \left\{ 1 + B_{x+1} \left[(x+1) \left(\frac{\mu}{\bar{\lambda}} \right) \sum_{y=1}^x \prod_{k=1}^y B_k - \left(1 + \sum_{y=1}^{x-1} \prod_{k=1}^y B_k \right) \right] \right\} \\
&= (W - c)\bar{\lambda}B(x) \left\{ B_{x+1} \left[(x+1) \left(\frac{\mu}{\bar{\lambda}} \right) \left\{ 1 + \sum_{y=1}^x \prod_{k=1}^y B_k \right\} - \left\{ 1 + \sum_{y=1}^{x-1} \prod_{k=1}^y B_k \right\} \right] \right\} \\
&= (W - c)\bar{\lambda}B(x)B_{x+1} \left[\left\{ (x+1) \left(\frac{\mu}{\bar{\lambda}} \right) - 1 \right\} A(x) + B(x) \right] \\
&= (W - c)\bar{\lambda}B(x+1) \left[\left\{ (x+1) \left(\frac{\mu}{\bar{\lambda}} \right) - 1 \right\} A(x) + B(x) \right]. \tag{20}
\end{aligned}$$

We now analyze the right side of (18), as follows:

$$\begin{aligned}
& h(x+1)B(x+1)A(x) + \sum_{n=0}^x hnB(n) \{A(x) - A(x+1)\} \\
&= B(x+1) \left\{ h(x+1)A(x) - \sum_{n=0}^x hnB(n) \right\} \\
&= B(x+1) \left\{ h(x+1)A(x) - h \left(\frac{\bar{\lambda}}{\mu} \right) A(x-1) \right\}. \tag{21}
\end{aligned}$$

Thus, equating (20) and (21) gives

$$\begin{aligned}
& (W - c)\bar{\lambda}B(x+1) \left[\left\{ (x+1) \left(\frac{\mu}{\bar{\lambda}} \right) - 1 \right\} A(x) + B(x) \right] = \\
& \quad hB(x+1) \left\{ (x+1)A(x) - \left(\frac{\bar{\lambda}}{\mu} \right) A(x-1) \right\},
\end{aligned}$$

or

$$W = c + \frac{h}{\bar{\lambda}} \left[\frac{(x+1)A(x) - \left(\frac{\bar{\lambda}}{\mu} \right) A(x-1)}{(x+1) \left(\frac{\mu}{\bar{\lambda}} \right) A(x) - A(x-1)} \right] = c + \frac{h}{\mu}.$$

Case 2: $x \geq s$

Let $\rho = \frac{\bar{\lambda}}{s\mu}$. For $x \geq s$, the equilibrium distribution for policy $\pi(x)$ is given by

$$p(x, k) = \begin{cases} \left(\frac{\bar{\lambda}}{\mu} \right)^k \frac{1}{k!} p(x, 0), & 0 \leq k \leq s, \\ \left(\frac{\bar{\lambda}}{\mu} \right)^s \frac{1}{s!} \rho^{k-s} p(x, 0), & s+1 \leq k \leq x, \end{cases}$$

where

$$p(x, 0)^{-1} = A(s) + \frac{B(s)\bar{\lambda}(1 - \rho^{x-s})}{s\mu - \bar{\lambda}}.$$

The defining equation of the index $W = W(x)$ is $C_{\pi(x)}(W) = C_{\pi(x+1)}(W)$, or

$$(W - c)\bar{\lambda} \{p(x, x) - p(x + 1, x + 1)\} = \sum_{n=0}^{x+1} hnp(x + 1, n) - \sum_{n=0}^x hnp(x, n). \quad (22)$$

Multiplying both sides of (22) by $p(x, 0)^{-1}p(x + 1, 0)^{-1}$ gives

$$\begin{aligned} & p(x, 0)^{-1}p(x + 1, 0)^{-1} [(W - c)\bar{\lambda} \{p(x, x) - p(x + 1, x + 1)\}] = \\ & p(x, 0)^{-1}p(x + 1, 0)^{-1} \left[\sum_{n=0}^{x+1} hnp(x + 1, n) - \sum_{n=0}^x hnp(x, n) \right]. \end{aligned} \quad (23)$$

We first develop the left side of (23) as follows:

$$\begin{aligned} & p(x, 0)^{-1}p(x + 1, 0)^{-1} [(W - c)\bar{\lambda} \{p(x, x) - p(x + 1, x + 1)\}] \\ & = (W - c)\bar{\lambda} \{B(s)\rho^{x-s}p(x + 1, 0)^{-1} - B(s)\rho^{x+1-s}p(x, 0)^{-1}\} \\ & = (W - c)\bar{\lambda}B(s)\rho^{x-s} \{p(x + 1, 0)^{-1} - \rho p(x, 0)^{-1}\} \\ & = (W - c)\bar{\lambda}B(s)\rho^{x-s}(1 - \rho) \left\{ A(s) + \frac{B(s)\bar{\lambda}}{s\mu - \bar{\lambda}} \right\}. \end{aligned} \quad (24)$$

We now analyze the right side of (23) by writing

$$\begin{aligned} & p(x, 0)^{-1}p(x + 1, 0)^{-1} \left[\sum_{n=0}^{x+1} hnp(x + 1, n) - \sum_{n=0}^x hnp(x, n) \right] \\ & = \sum_{n=0}^s h \left(\frac{\bar{\lambda}}{\mu} \right)^n \frac{1}{n!} n \left[A(s) + B(s) \left(\frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right) (1 - \rho^{x-s}) \right] \\ & \quad + \sum_{n=s+1}^x hB(s)\rho^{n-s}n \left[A(s) + B(s) \left(\frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right) (1 - \rho^{x-s}) \right] \\ & \quad + hB(s)\rho^{x+1-s}(x + 1) \left[A(s) + B(s) \left(\frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right) (1 - \rho^{x-s}) \right] \\ & \quad - \sum_{n=0}^s h \left(\frac{\bar{\lambda}}{\mu} \right)^n \frac{1}{n!} n \left[A(s) + B(s) \left(\frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right) (1 - \rho^{x+1-s}) \right] \\ & \quad - \sum_{n=s+1}^x hB(s)\rho^{n-s}n \left[A(s) + B(s) \left(\frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right) (1 - \rho^{x+1-s}) \right] \\ & = hB(s)\rho^{x+1-s}(x + 1) \left[A(s) + \frac{B(s)\bar{\lambda}}{s\mu - \bar{\lambda}} (1 - \rho^{x-s}) \right] \\ & \quad - hB(s)\rho^{x+1-s} \left[\left(\frac{\bar{\lambda}}{\mu} \right) \sum_{n=0}^{s-1} \left(\frac{\bar{\lambda}}{\mu} \right)^n \frac{1}{n!} + B(s)\rho \sum_{n=1}^{x-s} (s + n)\rho^{n-1} \right] \end{aligned} \quad (25)$$

$$\begin{aligned}
&= hB(s)\rho^{x+1-s}(x+1) \left[A(s) + \frac{B(s)\bar{\lambda}}{s\mu - \bar{\lambda}} (1 - \rho^{x-s}) \right] \\
&\quad - hB(s)\rho^{x+1-s} \left[\left(\frac{\bar{\lambda}}{\mu} \right) \{A(s) - B(s)\} \right. \\
&\quad\quad \left. + \frac{B(s)\bar{\lambda}}{s\mu - \bar{\lambda}} \left\{ s+1 + \frac{\bar{\lambda}}{s\mu - \bar{\lambda}} (1 - \rho^{x-s-1}) - x\rho^{x-s} \right\} \right] \\
&= hB(s)\rho^{x+1-s} \left[A(s) \left\{ x+1 - \frac{\bar{\lambda}}{\mu} \right\} \right. \\
&\quad \left. + \frac{B(s)\bar{\lambda}}{s\mu - \bar{\lambda}} \left\{ x + \rho^{x-s} \left(\frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right) - \frac{\bar{\lambda}}{\mu} - \frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right\} \right]. \tag{26}
\end{aligned}$$

Thus, equating (24) and (26) gives

$$W = c + \frac{h}{s\mu - \bar{\lambda}} \left[\frac{A(s) \left(x+1 - \frac{\bar{\lambda}}{\mu} \right) + B(s) \left(\frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right) \left\{ x + \left(\frac{\bar{\lambda}}{s\mu} \right)^{x-s} \left(\frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right) - \frac{\bar{\lambda}}{\mu} - \frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right\}}{A(s) + B(s) \left(\frac{\bar{\lambda}}{s\mu - \bar{\lambda}} \right)} \right].$$

This completes the proof for $\bar{\lambda} \neq s_j\mu_j$. The case $\bar{\lambda} = s_j\mu_j$ can either be dealt with similarly or by considering the limit $\bar{\lambda} \rightarrow s_j\mu_j$.

□

Comment: For the range $0 \leq x_j \leq s_j - 1$, the index is $c_j + h_j/\mu_j$, which is simply the expected cost incurred when a single job proceeds through the station unhindered by other queueing jobs, as is the case if the job is routed to a station with $x_j < s_j$. Asymptotically as $x_j \rightarrow \infty$, the index grows exponentially if $\bar{\lambda} > s_j\mu_j$, and it grows linearly if $\bar{\lambda} < s_j\mu_j$. If $\bar{\lambda} = s_j\mu_j$, the index grows quadratically for the range $x_j \geq s_j$.

In the case where none of the vendors can handle the entire demand stream alone (that is, $\bar{\lambda} > s_j\mu_j$ for all $j = 1, \dots, V$), all vendors will have the geometric index. The resulting index policy is less radical than it would at first appear, since choosing minimum index is equivalent to choosing minimum log index and the latter is asymptotically linear in the queue length. However, in cases where some vendors could handle the whole stream ($\bar{\lambda} < s_i\mu_i$, with linear index) while others could not ($\bar{\lambda} > s_j\mu_j$, with geometric index), the index policy would certainly make heavy use of the more capable vendors when the system becomes congested.

As in Section 4, in practice we modify the policy to account for the dynamically changing arrival rate. That is, we replace $\bar{\lambda}$ with $\lambda \left(K - \sum_{i=1}^V x_i \right)$ in the expression for $W_j(x_j)$. This is done in Section 6.

6 Simulation Study

The index policies derived in Sections 4 and 5 are easy to compute; however, there is no closed form expression for the expected cost of each index policy. Therefore, to evaluate the performance of the dynamic allocation policies in practice, we develop simulation models to estimate the average cost of implementing each index policy. In this section, we present the results of the simulation study. Our primary goals reflect the main objectives described in Section 1. In particular, our study demonstrates the following:

- (a) In all cases studied, the index-based dynamic routing heuristics developed in Sections 4 and 5 perform at least as well as optimal static allocation, and in most cases the size of the cost reduction is of practical significance.
- (b) In all cases studied, the dynamic routing heuristics are very close to optimal in the class of dynamic routing policies.
- (c) Our index-based heuristics consistently outperform two standard simple routing heuristics (JSQ and IO, described in Section 6.4). In many cases, the quality of performance of the latter was sufficiently weak as to render ineffective any cost comparison with static allocation procedures.

For each simulation in this section, we use 1,000 independent replications, and a duration of five years after a warm-up interval of two years. Our simulation programs were written in SIMSCRIPT II.5, and we used LABATCH.2 (Fishman [8]) to calculate 99% confidence intervals on the total cost to the manufacturer of following each policy.

In Section 6.1, we describe a measure of uniformity of the optimal static allocation from Opp et al. [25]. Using this measure of uniformity, in Section 6.2 we compare the two index policies to the optimal static allocation. In particular, we show that if the optimal static allocation is relatively uniform among the vendors, then dynamic allocation provides an opportunity for significant cost savings. Then, in Section 6.3, we compare the two index policies to the optimal dynamic routing policy for a limited set of examples with $K = 300$ and $V = 2$, and we show that the index policies are nearly optimal for these examples. Section 6.4 considers two additional simple routing heuristics, and compares both average case and worst case performance to the index policies.

6.1 Gini Coefficient of the Optimal Static Allocation

In preliminary experimentation with the policy improvement and restless bandit index policies, we noticed that the relative cost reduction from using dynamic allocation (rather than optimal static allocation) varied widely across problem instances. Furthermore, the reduction appeared to depend on the uniformity of the optimal static allocation. For example, as one might expect, problems in which all K items were allocated to one vendor in the optimal static allocation did not show much cost reduction when dynamic allocation was used. In fact, it was often the case for this type of problem that the dynamic allocation policy routed all failures to the same vendor used in the static allocation, effectively resulting in the same allocation policy as the optimal static allocation policy.

Therefore, to determine when dynamic allocation can provide a significant reduction in average costs, we calculate a measure of the uniformity of the optimal static allocation. The *Gini coefficient* is commonly used in economics as a measure of inequality in a population (Glasser [11], Sen [29]), and we apply it to our problem as a measure of the inequality of the allocation between vendors. Before giving an explicit expression for the Gini coefficient, we illustrate the concept with a small example.

Suppose a static allocation of 100 items to four vendors is given by $(15, 30, 10, 45)$; that is, vendor 1 is allocated 15 items, vendor 2 is allocated 30 items, vendor 3 is allocated 10 items, and vendor 4 is allocated 45 items. We sort the vendors according to their allocation, and we say that the lowest vendor receives 10% of the allocation, the lowest two vendors combined receive 25% of the allocation, and the lowest three vendors combined receive 55% of the allocation. Of course, the lowest four vendors combined receive 100% of the allocation. A Lorenz curve is a piecewise linear function that, in this case, plots the percent of allocation vs. the percent of vendors under this ordering. In the more common economics usage, the Lorenz curve plots the percent of income vs. the percent of households after ordering the households according to increasing income levels (Lorenz [22]). If all family incomes are equal, the Lorenz curve is a straight line connecting the points $(0, 0)$ and $(1, 1)$. Figure 1 shows the Lorenz curve and the line of perfect equality for the above allocation example.

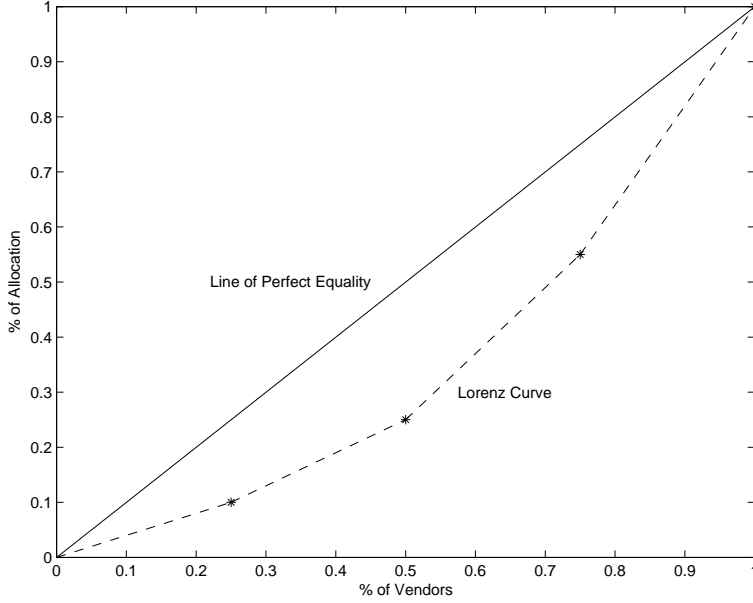


Figure 1: Lorenz curve and line of perfect equality

The Gini coefficient, G , is then calculated as the area between the line of perfect equality and the Lorenz curve, divided by the area beneath the line of perfect equality. In a perfectly equal allocation (for example, 25 items to each vendor when $K = 100$ and $V = 4$), the Lorenz curve will lie directly on the line of perfect equality, so $G = 0$. In the most unequal allocation (that is, all K items are allocated to one vendor), we have $G = (V - 1)/V$, since the Lorenz curve connects the points $(0, 0)$, $((V - 1)/V, 0)$, and $(1, 1)$. A smaller value of the Gini coefficient is generally taken to indicate a more uniform allocation among the vendors. For any allocation (k_1, \dots, k_V) of K items among V vendors, the Gini coefficient can be explicitly calculated using the following formula:

$$G = \frac{\sum_{i=j+1}^V \sum_{j=1}^V |k_i - k_j|}{KV}.$$

In the following section, we investigate the performance of the dynamic allocation index policies as a function of the Gini coefficient of the static allocation. We chose our examples from among the 81,920 trials used in Opp et al. [25]; these examples use $V = 4$ and a fixed failure rate of $\lambda = 1.2$ failures per item per year. Because of the computing time involved in each simulation, we only consider examples for $K = 100$ and $K = 1,000$. In Opp et al. [25], there are 20,480 trials for each of $K = 100$ and $K = 1,000$. We first computed the Gini coefficient of the optimal static allocation for each of these trials. The Gini coefficient ranged from 0.2 to 0.75 for the majority of these examples, with only a few instances having $G < 0.2$. Furthermore, there were far more examples with high values of G than with low values of G . Therefore, rather than randomly

selecting examples from all 20,480 trials for $K = 100$, we selected 15 examples with $\alpha \leq G < \alpha + 0.1$ for $\alpha \in \{0.2, 0.3, \dots, 0.7\}$ (where, for $\alpha = 0.7$, all 15 examples lie in the interval $[0.7, 0.75]$). We did the same thing for $K = 1,000$, for a total of 90 examples for each of these two values of K .

6.2 Comparison with the Optimal Static Allocation

In this section, we compare the two dynamic allocation index policies with the optimal static allocation from Opp et al. [25]. For the trials with $K = 100$, the estimated average cost of the dynamic policy using the policy improvement index was always less than the expected cost using optimal static allocation, and the reduction ranged from 0.06% to 18.12%. Furthermore, as was suggested by our initial experimentation, the size of the cost reduction from using dynamic allocation does depend on the uniformity of the optimal static allocation. Figure 2 shows the percent cost reduction using the policy improvement index policy for all 90 trials with $K = 100$, where the horizontal axis is the Gini coefficient of the optimal static allocation. The solid line has been fitted using linear regression, and the regression equation and R^2 value are given in the upper right corner of the plot. Clearly, the benefit from using dynamic allocation diminishes as the Gini coefficient, G , increases. This is to be expected, since a problem for which a static allocation policy allocates everything to one vendor will likely result in almost exactly the same policy under dynamic allocation. However, for the examples that resulted in a low value of G , the relative cost savings from using dynamic allocation instead of static allocation were much larger. For example, for the trials with $K = 100$ and G near 0.2, the cost reduction over optimal static allocation averaged around 15%. Figure 3 shows the results for the restless bandit index policy with $K = 100$. Because the two indices often result in nearly the same policy, the plots in Figures 2 and 3 are similar.

Figures 4 and 5 show the corresponding plots for the 90 trials using $K = 1,000$. These plots show the same downward trend as the plots for $K = 100$, but the relative cost reduction is a bit lower. The intercept of the linear regression equations for $K = 100$ are near 20, whereas the intercepts for $K = 1,000$ are just under 14. However, even though the relative cost reductions corresponding to $K = 1,000$ are smaller than those for $K = 100$, the actual cost reductions are much larger for the problems with $K = 1,000$, which have higher average cost than the problems with $K = 100$.

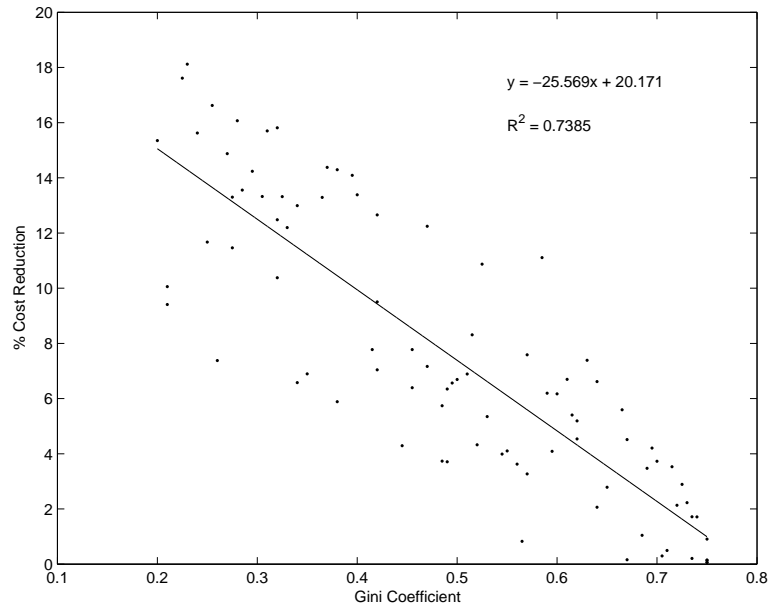


Figure 2: Percent cost reduction from using the policy improvement index policy rather than the optimal static allocation ($K = 100$)

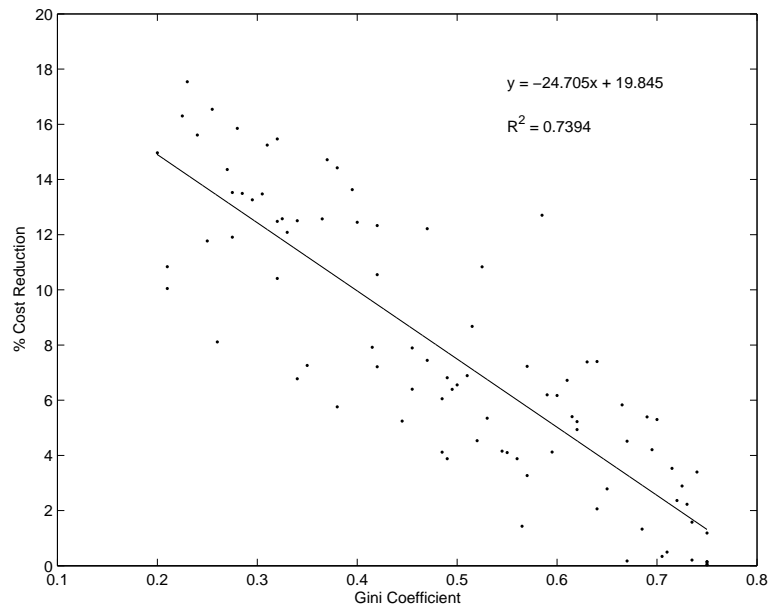


Figure 3: Percent cost reduction from using the restless bandit index policy rather than the optimal static allocation ($K = 100$)

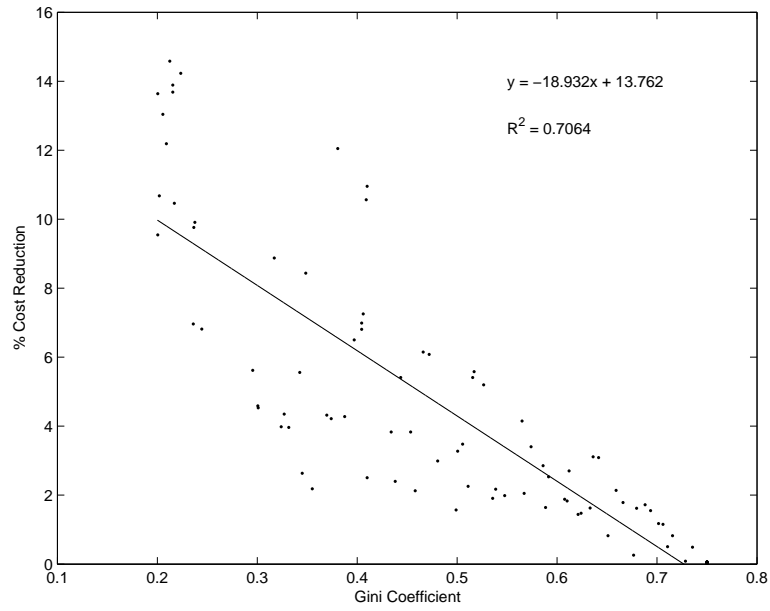


Figure 4: Percent cost reduction from using the policy improvement index policy rather than the optimal static allocation ($K = 1,000$)

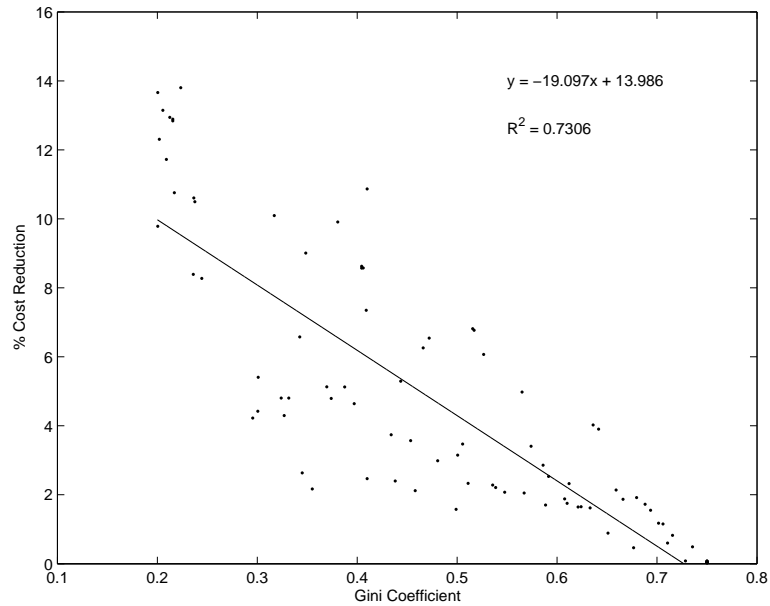


Figure 5: Percent cost reduction from using the restless bandit index policy rather than the optimal static allocation ($K = 1,000$)

Figures 2 through 5 seem to indicate that the percent cost reduction over static allocation is decreasing with K , which would imply that the benefit of using dynamic allocation would eventually disappear. However, due to the scaling of the data, the different values of K should be regarded as completely distinct problems from one another, rather than the same problem with a larger number of items to allocate. The reason for this is that for most problems, if the data is valid for both $K = 100$ and $K = 1,000$ (that is, if the stability condition $\sum_{i=1}^V s_i \mu_i > K\lambda$ holds for $K = 1,000$), then the results for $K = 100$ are usually uninteresting. In other words, if the total maximum service rate is enough to handle all 1,000 items, then it is often the case that if we only allocate 100 items among the vendors, the optimal allocation assigns everything to one vendor. In Opp et al. [25], the data were chosen such that all values of K would yield interesting results. As a result, the fact that the relative cost savings for $K = 1,000$ are smaller than those for $K = 100$ is an effect of using different data in the problems for $K = 100$ and $K = 1,000$, and is not a general result regarding increasing values of K .

To see that it is typically not the case that the relative cost savings decrease with K when all other problem parameters remain unchanged, consider the following example: $V = 4$, $\lambda = 1.2$, $\mu = (100, 100, 100, 100)$, $s = (2, 3, 4, 5)$, $c = (100, 110, 120, 130)$, and $h = (1000, 1000, 1000, 1000)$. This example is realistic in the sense that vendors with a higher maximum service rate ($s_i \mu_i$) are more attractive to a manufacturer, and hence charge a higher fixed cost per repair (c_i). We compute the costs of the optimal static allocation, the policy improvement index policy, and the restless bandit index policy as K ranges from 100 to 1,000.

The results are found in Table 1, where the columns labeled “Policy Improvement Index” and “Restless Bandit Index” give the percent cost reduction from using the corresponding index policy rather than optimal static allocation. Note that for both index policies, the percent cost reduction is generally increasing with K . For $K = 100$, the total failure rate ($K\lambda = 120$) is much lower than the total service capacity ($\sum_{i=1}^V s_i \mu_i = 1400$), so we would expect that not all vendors would be used in the optimal static allocation. In fact, only two vendors are used, and the Gini coefficient is rather high at $G = 0.6750$. As K increases, more vendors are needed, and the allocation is spread more evenly among the vendors. As a result, the Gini coefficient starts to decrease, and the percent cost reduction from using dynamic allocation starts to increase. As K increases, and the total failure rate becomes closer to the total service capacity, the manufacturer will start to favor the server with highest capacity (in this case, vendor 4). This is why the Gini coefficient increases after $K = 700$; however, the percent cost reduction for both index policies continues to increase.

Furthermore, the cost reductions are significant both in relative terms and in absolute terms, because as K increases, the average cost also increases. When $K = 100$, the policy improvement index policy provides a

1.302% cost reduction on an average cost of \$13,497 for the optimal static allocation, for a total savings of \$176. But for $K = 1,000$, the average cost of the optimal static allocation is \$162,700—more than ten times the cost for $K = 100$, due to the holding costs in the more congested system with 1,000 items to allocate. In this case, the cost reduction from using the policy improvement index policy is 5.560%, for a total savings of \$9,046.

Table 1: Percent cost reduction from using dynamic allocation rather than optimal static allocation, as a function of K

K	Gini	Policy Improvement Index	Restless Bandit Index
100	0.6750	1.302%	1.624%
200	0.5175	2.285%	2.302%
300	0.4250	2.984%	3.261%
400	0.3100	3.381%	3.329%
500	0.3470	3.599%	3.484%
600	0.1808	4.093%	3.846%
700	0.1150	4.204%	3.946%
800	0.1450	4.103%	4.092%
900	0.1739	4.436%	4.580%
1000	0.1825	5.560%	5.773%

6.3 Comparison with the Optimal Dynamic Routing Policy

For small values of K and two vendors (i.e., $V = 2$), we can compute the optimal dynamic routing policy using the optimality equations in Theorem 1. We consider 50 examples with $K = 300$, and we use value iteration to find the expected cost of the ϵ -optimal policy with $\epsilon = 0.0001$ (see Puterman [27]).

Table 2 summarizes the optimality gap for the 50 trials, using both the policy improvement index policy and the restless bandit index policy. For the 50 trials, the optimality gap for the policy improvement index policy ranged from 0% to 0.98%, with an average value of 0.26%. The optimality gap for the restless bandit index policy ranged from 0% to 0.25%, with an average value of 0.06%.

Table 2: Gap between the cost of dynamic allocation index policies and the cost of the optimal dynamic policy ($K = 300$)

	Policy Improvement Index	Restless Bandit Index
Min. gap	0%	0%
Max. gap	0.98%	0.25%
Mean gap	0.25%	0.06%

6.4 Heuristics

In this section we describe two very simple routing heuristics, and we compare the costs of these policies to the cost of both the policy-improvement index policy and the restless bandit index policy. The heuristics are as follows:

- **Join the Shortest Queue (JSQ):** An incoming item is sent to the vendor with the shortest queue length. If more than one vendor has minimal queue length, the item is sent to the vendor among them with the smallest value of the fixed cost c_j .
- **Individually Optimal (IO):** An incoming item is sent to the vendor for which the cost associated with that particular item alone is optimal; therefore, this heuristic myopically routes the incoming items. For each vendor j , the expected waiting time for an incoming item, EW_j , is calculated based on the service rate (μ_j), the number of servers (s_j), and the current state of the vendor (x_j). The IO heuristic then sends the item to the vendor that minimizes $c_j + h_j EW_j$.

Tables 3 and 4 summarize the gap between the average cost of using the routing heuristics (JSQ or IO) and the average cost of using the policy improvement or restless bandit index policy. These tables show that the JSQ and IO heuristics not only perform weakly on average, but there are problem instances for which the heuristics perform very poorly compared to the policy improvement and restless bandit index policies.

For instance, Table 3 shows the results for the join-the-shortest-queue heuristic. For the trials with $K = 100$, the average gap between JSQ and the policy improvement index policy is 7.874%, and the average gap between JSQ and the restless bandit index policy is 7.985%. For the trials with $K = 1,000$, the average gap between JSQ and the policy improvement index policy is 2.810%, and the average gap between JSQ and the restless bandit index policy is 2.946%. Again, recall that the data for $K = 100$ and $K = 1,000$ are for different problems, so a comparison of the gap between the two values of K is not meaningful.

Table 3: Gap between the cost of the JSQ heuristic and the cost of the dynamic allocation index policies

	Policy Improvement Index		Restless Bandit Index	
	$K = 100$	$K = 1,000$	$K = 100$	$K = 1,000$
Min. gap	-0.247%	-1.360%	-0.072%	-0.003%
Max. gap	41.043%	29.902%	41.376%	29.926%
Mean gap	7.874%	2.810%	7.985%	2.946%

Table 4 shows the results for the individually optimal heuristic. For the trials with $K = 100$, the average gap between IO and the policy improvement index policy is 4.164%, and the average gap between IO and the restless bandit index policy is 4.264%. For the trials with $K = 1,000$, the average gap between IO and the policy improvement index policy is 14.948%, and the average gap between IO and the restless bandit index policy is 15.140%.

Table 4: Gap between the cost of the IO heuristic and the cost of the dynamic allocation index policies

	Policy Improvement Index		Restless Bandit Index	
	$K = 100$	$K = 1,000$	$K = 100$	$K = 1,000$
Min. gap	-0.021%	-0.204%	-0.196%	-0.003%
Max. gap	16.193%	65.195%	15.894%	65.033%
Mean gap	4.164%	14.948%	4.264%	15.140%

7 Conclusions

In this paper, we consider the dynamic routing of items under warranty to alternative service vendors. Modeling the system as a continuous-time Markov decision process, we develop two separate index-type routing policies. Through a detailed computational study we demonstrate the following:

- (a) The index-based dynamic routing heuristics developed in Sections 4 and 5 can offer a significant cost reduction over the optimal static allocation, particularly in cases where the optimal static allocation is relatively uniform among the vendors.
- (b) The dynamic routing heuristics are very close to optimal in the class of dynamic routing policies; for the numerical cases studied, the policy-improvement heuristic was an average of 0.25% away from optimal, and the restless bandit heuristic was an average of 0.06% away from optimal.

- (c) The index-based heuristics consistently outperform the JSQ and IO routing heuristics, and there are problem instances for which the JSQ and IO routing heuristics perform very poorly.

For a manufacturer with annual warranty costs in tens of millions of dollars, improvements of this scale provide an opportunity to realize significant cost savings. Moreover, the closed form solutions for the indices (given in Theorems 2 and 3) are easy to calculate, resulting in policies that are easily manageable in practice.

Therefore, we make the following recommendation as to how to proceed when faced with a new practical problem. First, one should compute the optimal static allocation. If the Gini coefficient of the resulting allocation is very high (i.e., close to the maximum value of $(V - 1)/V$), then there is little to no advantage in using dynamic allocation. In this case, one should use static allocation to outsource the warranty repairs, thereby eliminating the cost of operating a central call facility. On the other hand, if the Gini coefficient is low, using either of the dynamic allocation index policies will likely produce significant cost reductions. Therefore, provided that information about the current state is available and the infrastructure supports dynamic routing decisions, one should implement dynamic routing using either the policy-improvement or restless bandit index policy.

References

- [1] P. S. Ansell, K. D. Glazebrook, J. Niño-Mora, and M. O’Keefe. Whittle’s index policy for a multi-class queueing system with convex holding costs. *Mathematical Methods of Operations Research*, 57(1):21–39, 2003.
- [2] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. 2*. Athena Scientific, Belmont, MA, 1995.
- [3] W. C. Cheng and R. R. Muntz. Optimal routing for closed queueing networks. *Performance Evaluation*, 13(1):3–15, 1991.
- [4] M. B. Combé and O. J. Boxma. Optimization of static traffic allocation policies. *Theoretical Computer Science*, 125(1):17–43, 1994.
- [5] L. W. Dowdy, D. L. Eager, K. D. Gordon, and L. V. Saxton. Throughput concavity and response time convexity. *Information Processing Letters*, 19(4):209–212, 1984.
- [6] M. E. Dyer and L. G. Proll. On the validity of marginal analysis for allocating servers in $M/M/c$ queues. *Management Science*, 23(9):1019–1022, 1977.
- [7] A. Ephremides, P. Varaiya, and J. Walrand. A simple dynamic routing problem. *IEEE Transactions on Automatic Control*, 25(4):690–693, 1980.
- [8] G. S. Fishman. *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer-Verlag, New York, 2001.
- [9] B. L. Fox. Discrete optimization via marginal analysis. *Management Science*, 13(3):210–216, 1966.
- [10] J. C. Gittins. *Multi-armed Bandit Allocation Indices*. John Wiley & Sons, New York, 1989.
- [11] G. J. Glasser. Variance formulas for the mean difference and coefficient of concentration. *Journal of the American Statistical Association*, 57(299):648–654, 1962.
- [12] K. D. Glazebrook, J. Niño-Mora, and P. S. Ansell. Index policies for a class of discounted restless bandit problems. *Advances in Applied Probability*, 34(4):754–774, 2002.
- [13] W. Grassmann. The convexity of the mean queue size of the $M/M/c$ queue with respect to the traffic intensity. *Journal of Applied Probability*, 20(4):916–919, 1983.

- [14] O. Gross. A class of discrete type minimization problems. Technical Report RM-1644, RAND Corp., 1956.
- [15] B. Hajek. Optimal control of two interacting service stations. *IEEE Transactions on Automatic Control*, 29(6):491–499, 1984.
- [16] A. Hordijk and J. A. Loeve. Optimal static customer routing in a closed queueing network. *Statistica Neerlandica*, 54(2):148–159, 2000.
- [17] T. Ibaraki and N. Katoh. *Resource Allocation Problems: Algorithmic Approaches*. MIT Press, Cambridge, MA, 1988.
- [18] K. R. Krishnan. Joining the right queue: a Markov decision rule. In *Proceedings of the 28th IEEE Conference on Decision and Control*, pages 1863–1868, 1987.
- [19] V. G. Kulkarni. *Modeling and Analysis of Stochastic Systems*. Chapman and Hall, New York, 1995.
- [20] H. L. Lee and M. A. Cohen. A note on the convexity of performance measures of $M/M/c$ queueing systems. *Journal of Applied Probability*, 20(4):920–923, 1983.
- [21] S. A. Lippman. Applying a new device in the optimization of exponential queueing systems. *Operations Research*, 23(4):687–710, 1975.
- [22] M. O. Lorenz. Methods for measuring the concentration of wealth. *Journal of the American Statistical Association*, 9:209–219, 1905.
- [23] H. Luss and S. K. Gupta. Allocation of effort resources among competitive activities. *Operations Research*, 23(2):360–366, 1975.
- [24] J. Niño-Mora. Dynamic allocation indices for restless projects and queueing admission control: a polyhedral approach. *Mathematical Programming*, 93:361–413, 2002.
- [25] M. Opp, I. Adan, V. G. Kulkarni, and J. M. Swaminathan. Outsourcing warranty repairs: Static allocation. Technical Report UNC/OR TR-03-1, Department of Operations Research, UNC-Chapel Hill, April 2003.
- [26] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of optimal queueing network control. *Mathematics of Operations Research*, 24(2):293–305, 1999.
- [27] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, 1994.

- [28] A. J. Rolfe. A note on marginal allocation in multiple-server service systems. *Management Science*, 17(9):656–658, 1971.
- [29] A. Sen. *On Economic Inequality*. Clarendon Press, Oxford, England, 1973.
- [30] J. G. Shanthikumar and D. D. Yao. Optimal server allocation in a system of multi-server stations. *Management Science*, 33(9):1173–1180, 1987.
- [31] J. G. Shanthikumar and D. D. Yao. On server allocation in multiple center manufacturing systems. *Operations Research*, 36(2):333–342, 1988.
- [32] S. Stidham and R. Weber. A survey of Markov decision models for control of networks of queues. *Queueing Systems*, 13(1–3):291–314, 1993.
- [33] H. C. Tijms. *Stochastic Models: An Algorithmic Approach*. John Wiley & Sons, New York, 1994.
- [34] R. R. Weber and G. Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990.
- [35] P. Whittle. Restless bandits: activity allocation in a changing world. *Journal of Applied Probability*, 25A:287–298, 1988.
- [36] P. Whittle. *Optimal Control: Basics and Beyond*. John Wiley & Sons, New York, 1996.
- [37] P. H. Zipkin. Simple ranking methods for allocation of one resource. *Management Science*, 26(1):34–43, 1980.