European Journal of Operational Research 217 (2012) 554–559

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Decision Support

Maximizing throughput in finite-source parallel queue systems

Mohammad Delasay, Bora Kolfal, Armann Ingolfsson *

School of Business, University of Alberta, Edmonton, AB T6G 2R6, Canada

ARTICLE INFO

Article history:
Received 2 January 2011
Accepted 26 September 2011
Available online 8 October 2011

Keywords:
Markov processes
Queueing
Routing to parallel queues
Dispatching systems
Markov decision processes

ABSTRACT

Motivated by the dispatching of trucks to shovels in surface mines, we study optimal routing in a Markovian finite-source, multi-server queueing system with heterogeneous servers, each with a separate queue. We formulate the problem of routing customers to servers to maximize the system throughput as a Markov Decision Process. When the servers are homogeneous, we demonstrate that the Shortest Queue policy is optimal, and when the servers are heterogeneous, we partially characterize the optimal policy and present a near-optimal and simple-to-implement policy. We use the model to illustrate the substantial benefits of pooling, by comparing it to the permanent assignment of customers to servers.

1. Introduction

In this paper, we formulate a Markov Decision Process (MDP) model for the routing of customers from a finite population to a set of parallel queues with the objective of maximizing average throughput (number of customers served) per time unit. The left panel of Fig. 1 illustrates the system. Customers arrive from the population of size N to a routing point, where they are routed (assigned) to one of s parallel servers. Each server has a separate queue, and customers cannot jockey between queues. Service times are independent and exponentially distributed with parameter μ_i for server i, $i = 1,\ldots,s$. After completing service, customers return to the population, where they spend independent and exponentially distributed amounts of time (which we refer to as *backcycle times*) with parameter λ before returning to the routing point.

Our primary motivating example is the routing of trucks (customers) to shovels (servers) in a surface mine (for example, in the Alberta oilsands). In oilsands mines, the size of the extraction plant determines the throughput rate required to keep the plant running. Given a required throughput rate, one can view the planning problem as minimizing the cost of shovels and trucks needed to achieve that throughput rate. We frame the problem as one of routing trucks to shovels so as to maximize throughput, given a set of shovels and trucks. Our model could be solved with different sets of shovels and with different truck fleet sizes, in order to determine the least costly configuration that achieves the required throughput. The right panel of Fig. 1 illustrates a surface mining operation. Trucks circulate between shovels, where they are filled with ore; a dump location; and a dispatch point, where they are routed to one of the shovels. Ta et al. (2010) provide further information about oilsands mines and discuss how one can assign trucks to shovels in a static fashion so as to minimize the number of trucks while achieving the required throughput and satisfying other constraints. We use our model to illustrate the substantial pooling benefits of real-time truck routing over the static assignment of trucks to shovels. In one numerical example, we estimate that pooling could save $12.8 million per year.

Our model and results could also be relevant, for example, to the assignment of waste collection trucks to routes or the assignment of airplanes to repair facilities for preventive maintenance. A distinguishing feature of both of these examples, as well as of the routing of trucks in surface mines, is that jockeying (switching from one queue to another) involves traveling a substantial distance and is therefore unlikely to occur. We assume no jockeying in our model.

Optimal customer routing polices have been studied extensively; see Stidham and Weber (1993) for an overview. We classify the related literature based on whether the customer population is finite or infinite and based on the objective function. Most research on customer routing assumes an infinite customer population. Models with a finite customer population have mainly been studied in the context of machine interference. Two widely used objectives are to maximize throughput (customers served) and to minimize holding cost (which translates to minimizing the average number of customers waiting or being served if the holding costs are homogeneous).

Winston (1977) and Hordijk and Koole (1990) seek to maximize throughput in an infinite customer population system with parallel

* Corresponding author. Tel.: +1 780 492 7982; fax: +1 780 492 3325.
E-mail addresses: delasays@ualberta.ca (M. Delasay), bora.kolfal@ualberta.ca (B. Kolfal), armann.ingolfsson@ualberta.ca (A. Ingolfsson).

0377-2217/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2011.09.041

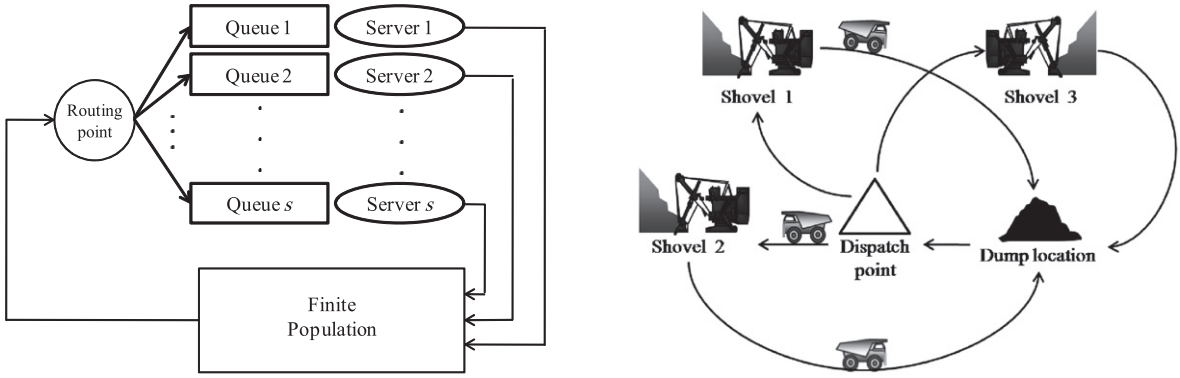

Fig. 1. General finite population routing to parallel queues problem and its application in surface mining.

servers, each with a separate queue, where service times are exponentially distributed with an equal mean. Winston's model assumes a Poisson arrival process, while Hordijk and Koole allow a general arrival process. Weber (1978) extends Winston's analysis to allow for service time distributions with a non-decreasing hazard rate. All three studies demonstrate that the Shortest Queue (SQ) policy is optimal. However, the arrival processes are assumed to be exogenous (independent of the number of customers in service or waiting to get service) in all three studies, which rules out an arrival process from a finite population.

Koole et al. (1999) show that in homogeneous infinite customer population systems, the SQ policy is optimal with respect to various cost functions, for systems with two identical servers and a broad range of service-time distributions. In infinite customer population models with homogeneous servers, minimizing the average number of customers waiting or receiving service is often equivalent to minimizing both the average waiting time and the average workload (Koole, 2005).

For heterogeneous servers, Hordijk and Koole (1992) partially characterize optimal routing policies to more than two parallel queues with exponential servers, by proving that routing to the faster server, when that server has a shorter queue, minimizes expected cost. Xu and Zhao (1996) extend the study of two heterogeneous servers to permit jockeying between the queues, and they characterize the routing policy that minimizes the expected holding and jockeying cost. Larsen and Agrawala (1983), Xu et al. (1992), and Koyanagi and Kawai (1995) propose optimal threshold routing policies for two-server systems with respect to various cost functions. With these policies, customers are routed to the fast server (regardless of the status of the slow server) until a certain threshold value for the fast-server queue size, at which point a customer is removed from that queue and sent to the slow server.

Much of the study of finite-population queueing systems has focused on a machine repair context. The prescriptive analysis of the multi-server "machine repairman problem" mainly focuses on two types of decisions: (1) the number of servers, the servers' service rates, and the machines' failure rates (Albright, 1980; Ching, 2001; Ke and Wang, 1999; Wartenhorst, 1995), and (2) the sequencing of repair service on failed machines (Frostig, 1993).

Few papers address the optimal assignment of repair people to failed machines. Righter (1996) studies the routing of failed machines with arbitrary arrivals to heterogeneous exponential servers and shows that sending the next job to the fastest available server stochastically maximizes the number of service completions. The main difference between Righter's model and our model is that in Righter's model failed machines wait in a common buffer, whereas in our model the servers have separate queues, and jockeying is not permitted. (In the surface mining context, jockeying corresponds to a truck traveling from one shovel to another, which is unlikely to happen.)

Perhaps the studies closest to ours are those of Goheen (1977) and Cinlar (1972), who address the problem of routing failed machines to repair stations (with separate buffers for each station, as in our model) so as to minimize the long-run average cost (as opposed to throughput, as in our model). Cinlar (1972) assumes exponential repair times, whereas Goheen (1977) assumes Erlang repair times. Goheen and Cinlar demonstrate that an optimal policy exists and can be found by solving either a linear (Cinlar, 1972) or nonlinear (Goheen, 1977) program, but they do not analyze the structure of those policies or provide computational results.

We extend the study of customer routing to parallel queues to finite source populations. We partially characterize the optimal routing policy for two-server systems and demonstrate that the SQ policy is optimal for an arbitrary number of homogeneous exponential servers. When the servers are heterogeneous, the optimal routing policy is complex. We propose an easy-to-implement and near-optimal heuristic policy for systems with an arbitrary number of heterogeneous servers. The policy begins by eliminating servers that are so slow that the optimal policy rarely or never uses them and then uses an easily computed index policy for the remaining servers. Our numerical results show that our policy performs extremely well for a wide range of model parameters.

Section 2 presents the MDP model and our assumptions. Section 3 presents several structural properties of the optimal routing policy for two-server systems and proves that the optimal policy for systems with an arbitrary number of homogeneous servers is SQ. Section 4 describes our near-optimal heuristic policy for systems with arbitrarily many heterogeneous servers. Finally, Section 5 numerically evaluates the performance of our proposed heuristic policy, illustrates how server utilization depends on server speed under the optimal policy, and illustrates the benefits of pooling.

2. Model formulation

Let $S = \{1,\ldots,s\}$ be the set of servers, let $n_i \in \{0,\ldots,N\}$ be the number of customers waiting or being served by server $i, i \in S$, and let $\mathbf{n} = (n_1,\ldots,n_s)$. The state space of our MDP model is $\Omega = \{\mathbf{n} : \mathbf{n} \in Z^n, \sum_{i\in S} n_i \leqslant N\}$. There are two types of decision epochs: (1) service completion by server i, where one decides whether to begin serving the next customer (if $n_i > 0$) or to idle and (2) arrival of a customer to the routing point, where one decides where to route the arriving customer.

Let I_R be an indicator function for event R, and let $\Lambda = \sum_{i=1}^{s} \mu_i + N\lambda$. Recall that $1/\lambda$ is the average time that customers spend in the population before returning to the routing point, and

$1/\mu_i$ is the average service time for server i. Using uniformization (Lippman, 1975), we express the MDP optimality equation as

$$\frac{g}{\Lambda}+v(\mathbf{n})=\sum_{i=1}^{s}\frac{\mu_i}{\Lambda}\max\{I_{n_i\geqslant 1}+v(\mathbf{n}-\mathbf{e}_i\times I_{n_i\geqslant 1}),v(\mathbf{n})\}+\frac{(N-\sum_{i=1}^{s}n_i)\lambda}{\Lambda}\max_{i\in S}\{v(\mathbf{n}+\mathbf{e}_i)\}+\frac{\lambda\sum_{i=1}^{s}n_i}{\Lambda}v(\mathbf{n}),\qquad(1)$$

where v is the optimal value function, g is the optimal throughput per time unit, and $\mathbf{n}\pm\mathbf{e}_i=(n_1,\ldots,n_i\pm 1,\ldots,n_s)$. The first term of the right-hand side represents decisions made upon a service completion: to be idle or to serve the next customer. If server i serves a customer, the throughput increases by one unit, and n_i decreases by one. If the server idles, then the throughput and n_i stay the same. The second term on the right-hand side presents the routing decision, where n_i increases by one if the arriving customer is routed to server i, while the number of customers in the other queues does not change.

Service times and backcycle times are typically not exponentially distributed in real surface mines. These times are often well-approximated by Erlang distributions, both in oil sands mines (Ta et al., 2010) and other similar systems (Carmichael, 1986a; Carmichael, 1986b). We expect our model to be relatively robust to the assumption of exponential backcycle times, because the steady state probabilities of multi-server finite-source queues with i.i.d. exponential service times are insensitive to the shape of the backcycle time distribution (Gross et al., 2008). Carmichael (1986b) argues that this invariance result holds approximately even if the service time distribution is not exponential. We discuss the assumption of exponential service times in Section 6.

3. Structural properties of the optimal policy

In this section, we partially characterize the optimal routing policy for systems with two heterogeneous servers. The proofs of all results in this section are in the Appendix. We prove that unforced idling (idling when there is at least one customer in the queue) is never optimal. We also show that the optimal routing policy for queue i is monotone in n_i. For systems with an arbitrary number of homogeneous servers, we show that the SQ policy is optimal.

Let D_j be the first difference operator, defined as $D_jv(\mathbf{n})=v(\mathbf{n}+\mathbf{e}_j)-v(\mathbf{n})$ for a function v of a vector $\mathbf{n}$ of two integer variables, and define the second difference operators as $D_{ii}=D_iD_i$ and $D_{ij}=D_{ji}=D_iD_j$. We use "increasing" and "decreasing" in the weak sense of "non-decreasing" and "non-increasing" throughout. Let Ψ be the set of functions v defined on the state space Ω that have the following properties:

- **P1.** *Submodularity* ($D_{ij}v\leqslant 0;\forall i,j\in S,i\neq j$): D_iv is decreasing in n_j, and D_jv is decreasing in n_i.
- **P2.** *Diagonal submissiveness* ($D_{ii}v\leqslant D_{ij}v;\forall i,j\in S,i\neq j$): D_jv-D_iv is increasing in n_i and decreasing in n_j.
- **P3.** *Concavity* ($D_{ii}v\leqslant 0;\forall i\in S$): Properties **P1** and **P2** together imply concavity. If v is concave, then D_iv is decreasing in n_i.
- **P4.** *Upper boundedness* ($D_iv\leqslant 1;\forall i\in S$).

We define the operators T_{μ_i}, $i\in S$, and T_λ as follows:

$$T_{\mu_i}v(\mathbf{n})=\max\{I_{n_i\geqslant 1}+v(\mathbf{n}-\mathbf{e}_i\times I_{n_i\geqslant 1}),v(\mathbf{n})\}\quad i\in S,\qquad(2)$$

$$T_\lambda v(\mathbf{n})=\max_{i\in S}\{v(\mathbf{n}+\mathbf{e}_i)\}.\qquad(3)$$

Setting $\Lambda=1$, we define operator T to represent the right hand side of (1) as

$$Tv(\mathbf{n})=\sum_{i=1}^{s}\mu_iT_{\mu_i}v(\mathbf{n})+\left(N-\sum_{i=1}^{s}n_i\right)\lambda T_\lambda v(\mathbf{n})+\sum_{i=1}^{s}n_i\lambda v(\mathbf{n}).\qquad(4)$$

Lemma 1 shows that the properties **P1**–**P4** are preserved under the operator T and the optimal value function $v\in\Psi$:

Lemma 1. *Let τ be a real valued function defined on Ω. If $\tau\in\Psi$, then (1) $T_{\mu_i}\tau\in\Psi$, $i\in S$, (2) $T_\lambda\tau\in\Psi$, and (3) $T\tau\in\Psi$. Furthermore, the optimal value function $v\in\Psi$.*

Lemma 1 enables us to prove the following two theorems:

Theorem 1. *If a queue is nonempty, then it is suboptimal for its server to be idle.*

Theorem 2. *If routing to server i is optimal in state $\mathbf{n}$, then routing to server i is also optimal in states $\mathbf{n}-\mathbf{e}_i$ (when $n_i>0$) and $\mathbf{n}+\mathbf{e}_j$ (when $n_j<N$, $j\neq i$).*

Theorem 1 simplifies the optimality Eq. (1) to

$$g+v(\mathbf{n})=\sum_{i=1}^{s}\mu_i(I_{n_i\geqslant 1}+v(\mathbf{n}-\mathbf{e}_i\times I_{n_i\geqslant 1}))+\left(N-\sum_{i=1}^{s}n_i\right)\lambda T_\lambda v(\mathbf{n})+\lambda\sum_{i=1}^{s}n_iv(\mathbf{n}).\qquad(5)$$

Based on Theorem 2, given the optimal decision in one state, we can deduce the optimal routing decision for several other states. For example, for the two-server system shown in Fig. 2, routing to server 2 is optimal in state (1,2). Therefore, based on Theorem 2, it is also optimal to route to server 2 for all the states to the left of or below state (1,2).

Lemma 2 allows us to characterize the optimal policy further, in Theorem 3.

Lemma 2. *For $i,j\in\{1,2\}$,*

(i) *If $n_i\geqslant n_j>0$, $\mu_j\geqslant\mu_i$, and $v(\mathbf{n}+\mathbf{e}_j)\geqslant v(\mathbf{n}+\mathbf{e}_i)$, then $v(\mathbf{n}-\mathbf{e}_i+\mathbf{e}_j)\geqslant v(\mathbf{n}+\mathbf{e}_i-\mathbf{e}_j)$.*
(ii) *If $n_i=n_j$, and $\mu_j\geqslant\mu_i$, then $v(\mathbf{n}+\mathbf{e}_j)\geqslant v(\mathbf{n}+\mathbf{e}_i)$.*

Theorem 3. *When $n_1=n_2$, it is optimal to route arriving customers to the faster server.*

Theorems 2 and 3 together allow us to determine the optimal routing decision for more than half of all states for two-server heterogeneous systems (as illustrated in Fig. 2): Theorem 3 specifies the optimal decision for states on the diagonal $n_1=n_2$, and Theorem 2 specifies the optimal decision either for all states above or all states below the diagonal.

For homogeneous s-server systems ($\mu_i=\mu$ for $i\in S$, $s\geqslant 2$), the SQ policy is optimal, as shown in the following Theorem.

Theorem 4. *Routing to the Shortest Queue policy is optimal for systems with an arbitrary number of homogeneous servers.*

n_1 \ n_2	0	1	2	3	4	5
0	2	1	1	1	1	1
1	2	2	2	1	1	
2	2	2	2	2		
3	2	2	2			
4	2	2				
5	2					

Fig. 2. Optimal routing policy for a heterogeneous system with $N=6$, $\lambda=2$, $\mu_1=2$, and $\mu_2=4$ (1: route to server 1, 2: route to server 2).

In the proof of Theorem 4 in the Appendix, we show that SQ stochastically maximizes throughput, which implies that SQ maximizes expected throughput.

4. Two-stage near-optimal policy for general systems

The complex structure of the optimal policy, even for two-server systems, motivated us to develop an easy-to-use, near-optimal heuristic policy. Implementing the optimal policy would require the use of an s-dimensional lookup table (for a system with s servers), as well as convincing the system operator that the resulting policy is a sensible one. Thus, we seek a policy that is easily justified, in addition to being simple to implement and close to optimal. Our proposed policy begins with Server Elimination (SE) and then uses a Modified Least Remaining Workload (MLRW) index policy for the remaining servers.

The SE stage was motivated by our observation, based on numerical experiments, that if a server is sufficiently slow (μ_i is sufficiently small), then the optimal policy never routes to that server. Based on extensive numerical experiments with two-server systems, we found that removing server i was near-optimal if the following inequality was satisfied:

$$\mu_j \geqslant N\mu_i + \lambda(N/2 - 1), \tag{6}$$

where μ_i, μ_j, and λ are not normalized. We developed this expression by considering two special cases of two-server systems. Suppose server 1 is the fast one, that is, $\mu_1 > \mu_2$. First, consider a system where λ is small enough that $\lambda N \approx 0$, suppose that $(n_1,n_2) = (N-1,0)$, and suppose that the remaining customer arrives to the routing point while the system is in this state. The expected time until that customer completes service is minimized by routing to the fast server, if the condition $\mu_1 \geqslant N\mu_2$ holds. In states where $n_1 < N-1$, there would be even greater reason to route an arriving customer to the fast server. Therefore, if $\mu_1 \geqslant N\mu_2$, then it will never be beneficial to route a customer to the slow server, and therefore nothing is lost by removing the slow server.

Second, suppose that the term $N\lambda$ is not close to zero. The risk that we take when routing to the slow server is that the fast server becomes idle before the slow server completes serving the customer that was routed to it. As $N\lambda$ increases, this risk decreases, because the probability that another customer arrives to the routing point before the slow server completes its service increases. Therefore, the threshold rate that the fast server must exceed in order to remove the slow server should increase with $N\lambda$. The term we used, $\lambda(N/2-1)$, was chosen because, in the special case when $N = 2$, as long as $\mu_1 \geqslant 2\mu_2$, routing to the fast server will minimize the expected time until the routed customer completes service, regardless of the magnitude of λ.

For systems with more than two servers, we check whether (6) holds for the fastest server (j) and the slowest server (i), and, if so, we eliminate server i. We continue this procedure recursively, until no more servers can be eliminated.

To motivate the MLRW policy, consider the following two greedy strategies, which focus only on the customer that is currently at the routing point and ignore all future arrivals to the routing point.

- Least Time to Complete Service (LTCS): Route to the queue i^* with the least expected time until service of the current customer is completed, that is

$$i^* = \arg\min_{i\in S}\left\{\frac{n_i+1}{\mu_i}\right\}. \tag{7}$$

- Least Remaining Workload (LRW): Route to the queue i^* with the least expected time until all currently assigned customers have been served, that is

$$i^* = \arg\min_{i\in S}\left\{\frac{n_i}{\mu_i}\right\}. \tag{8}$$

Neely et al. (2002) introduced LTCS and LRW in the context of satellite and wireless networks and referred to them as a greedy strategy (LTCS) and a work-conserving strategy (LRW), respectively.

Our MLRW policy chooses the server to route to by minimizing a quantity that is between the expected time to complete service and the expected remaining workload, and it incorporates a term that depends on the parameters N and λ of the finite source population. Specifically, the MLRW policy routes to the queue i^*, where

$$i^* = \arg\min_{i\in S}\left\{\frac{n_i}{\mu_i} + \frac{N - \sum_{j\in S} n_j}{N\mu_i}\right\}. \tag{9}$$

The first term, n_i/μ_i, is the expected remaining workload for server i. The second term,

$$0 < \frac{N - \sum_{j\in S} n_j}{N\mu_i} \leqslant \frac{1}{\mu_i},$$

equals the fraction of customers in the population times the average service time for server i. When the population size approaches infinity, or when the number of customers being served or waiting to be served ($\sum_j n_j$) approaches zero, this term approaches $1/\mu_i$, and the MLRW policy approaches the LTCS policy. On the other hand, when most of the customers are being served or waiting to be served, this term approaches zero, and the MLRW policy approaches the LRW policy. If the servers are homogeneous, then the second term becomes identical for all servers, and the MLRW policy reduces to the SQ policy.

In the next section, we provide evidence that MLRW performs significantly better than LTCS and LRW and, furthermore, that our two-stage policy (SE + MLRW) is near-optimal.

5. Numerical analysis

We designed an extensive test suite of 544 problems with two, three, or five servers, to compare the performance of the LTCS, LRW, MLRW, SE + MLRW, and optimal routing policies. Table 1 lists the parameters we used in the test suite, which included both problems with homogeneous and heterogeneous servers.

We computed the performance error percentage as the throughput difference between the SE + MLRW policy and the optimal policy. Table 2 summarizes the descriptive statistics of the performance error percentage for the two, three, and five-server test problems and for the whole test suite. Fig. 3 depicts the survivor function of the performance error percentage for each of the problem groups.

The results show that the SE + MLRW policy performs extremely well and provides the optimal solution in 49.8% percent of the test problems. Even when the problem size (the population size or the number of servers) increases, the SE + MLRW policy still provides near-optimal performance. For the test suite, the mean, standard deviation, and maximum error percentages are only 0.15%, 0.28%, and 1.62%, respectively.

Tables 3 and 4 summarize the percent increase in throughput from using the MLRW policy over the LRW and LTCS policies, respectively. Although there are test cases in which the LRW or the LTCS policy performs better than the MLRW policy (by up to 1.51%), MLRW outperforms these policies on average and, in some cases, by a large amount. In particular, the LRW policy performs quite poorly for the 5-server test cases.

Fig. 4 shows how server utilization increases with population size for a three-server system with $\mu_1 = 1$, $\mu_2 = 2$, and $\mu_3 = 4$ under

Table 1
Parameters of the test problems.

s	No. of problems	N	λ	μ_1	μ_2/μ_1	μ_3/μ_2	μ_4/μ_3	μ_5/μ_4
2	112	{3,6,...,21}	1	{0.5,1,2,4}	{1,1.5,2,4}	–	–	–
3	252	{3,6,...,21}	1	{0.5,1,2,4}	{1,2,4}	{1,2,4}	–	–
5	180	{3,6,...,15}	1	{0.5,1,2,4}	{1,2,4}	{1}	{1}	{1,2,4}

Table 2
Descriptive statistics for the performance error percentage (Optimal vs. SE + MLRW policy).

s	Mean (%)	Standard deviation (%)	% Optimum	Maximum (%)
2	0.05	0.13	58.9	0.71
3	0.18	0.30	42.1	1.44
5	0.18	0.15	56.1	1.62
Combined	0.15	0.28	49.8	1.62

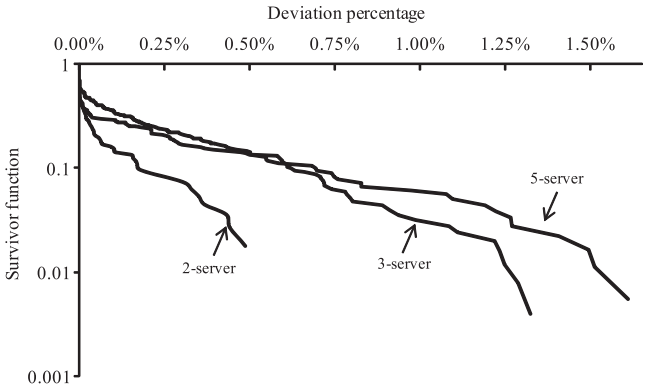

Fig. 3. Survivor function of the performance error percentage.

Table 3
Descriptive statistics for the throughput improvement of the MLRW policy over the LRW policy.

s	Mean (%)	Standard deviation (%)	% Minimum	Maximum (%)
2	0.14	0.01	−0.60	3.80
3	0.64	1.54	−1.33	9.44
5	7.34	8.16	−0.36	44.93
Combined	2.57	5.56	−1.33	44.93

Table 4
Descriptive statistics for the throughput improvement of the MLRW policy over the LTCS policy.

s	Mean (%)	Standard deviation (%)	% Minimum	Maximum (%)
2	0.11	0.28	−0.33	1.29
3	0.25	0.63	−0.34	3.74
5	0.15	0.72	−1.51	5.08
Combined	0.19	0.01	−1.51	5.08

the optimal policy, illustrating that servers with higher service rates are utilized more—a pattern that we observed in all of the test problems.

We finish by illustrating the benefits of customer pooling that are achieved in the system we model, by dynamically routing customers to servers based on the state of the system. In contrast, in an unpooled system, each customer is permanently assigned to one of the servers. To simplify the comparison, we consider systems with homogeneous servers and an even number of customers. We also compare to a single-queue system, which provides

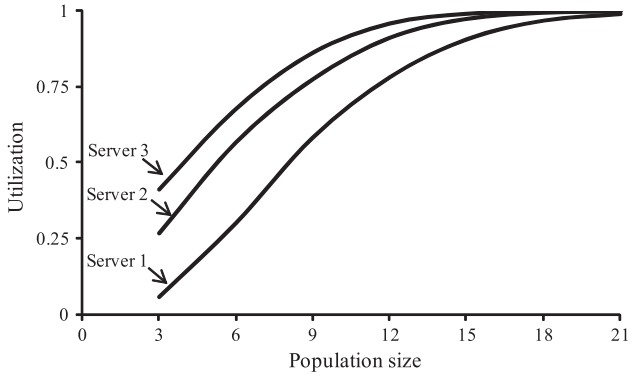

Fig. 4. Server utilization for a three-server system with $\mu_1 = 1$, $\mu_2 = 2$, and $\mu_3 = 4$.

an upper bound on the benefits that can be achieved from pooling. The left panel of Fig. 5 compares the throughput of pooled two-server systems to unpooled and single-queue systems. Interpreting Fig. 5 (left) in the context of surface mining, the pooled system requires 12 trucks, while the unpooled system needs 20 trucks to obtain a target throughput of 7 units, and the pooled system almost achieves the single-queue upper bound (but note that the single-queue system is unrealistic in a surface mining context).

Being able to reduce the number of trucks by eight translates into considerable cost savings. A typical 360-ton haul truck costs $5 to 6 million (Oil Sands Discovery Centre, 2011). Roman and Daneshmend (2000) estimated the annual operating and maintenance cost of a 360-ton truck to be $870,000 in the year 2000. Assuming 20% inflation from 2000 to 2011 and assuming a useful life of 10 years, we estimate the sum of depreciation, operating, and maintenance costs for one 360-ton haul truck to be approximately $1.6 million per year, which translates to $12.8 million in savings per year, if the number of trucks is reduced by eight.

The benefit of pooling is greater for lightly utilized systems. Intuitively, in a lightly utilized unpooled system, situations in which one server is idle while another server has waiting customers will be relatively common. The balancing effect of pooling reduces the frequency of such situations. Fig. 5 (left) illustrates that, as the population size increases (and, therefore, server utilization increases), the benefit of pooling decreases; the system throughput for the unpooled, pooled, and single-queue systems approaches the limit imposed by the combined capacities of the servers. The right panel of Fig. 5 illustrates this effect further, showing how the benefit of pooling (or, of having a single queue) increases when the service rate in a system with two homogeneous servers increases (and, therefore, server utilization decreases).

6. Further research

Our analysis constitutes a first step towards investigating the structure of optimal truck dispatching policies for surface mines. We demonstrate that such policies could result in substantial savings, compared to the static assignment of trucks to shovels, but much research remains to be done to better understand shovel-truck systems. From an applications perspective, we believe that

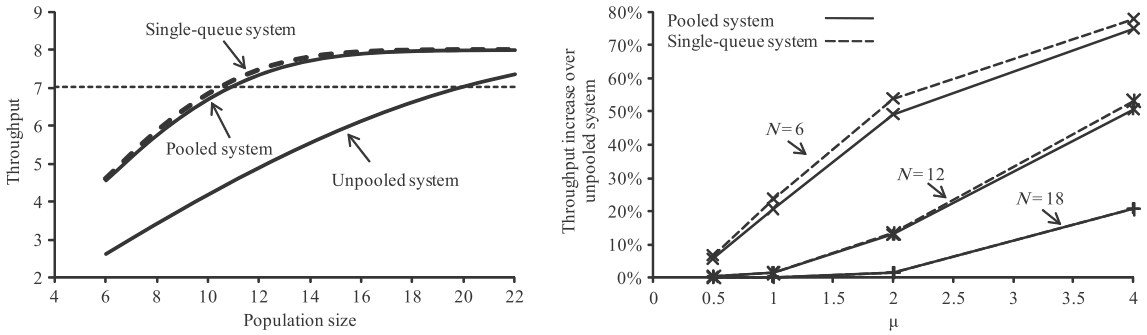

Fig. 5. Pooling strategy effect on throughput. (Left: $\lambda = 1$ and $\mu_1 = \mu_2 = 4$, right: $\lambda = 1$ and $\mu_1 = \mu_2 = \mu$.)

a natural next step would be to relax an assumption that is implicit in our model, namely, that a truck enters the queue for the shovel it is assigned to immediately after it passes the routing point. In reality, after being routed, trucks must travel to their assigned shovel. We expect that the travel times will vary (depending, for example, on weather) and that trucks could break down while traveling to a shovel. These considerations suggest a model with an infinite-server node between the routing point and each of the shovel queues, with average service times at the infinite-server nodes that represent travel times. We leave the investigation of such a model for future research.

The sensitivity of our model to the assumption of exponential service times should also be investigated. Weber (1978) demonstrated that the SQ policy for systems with homogeneous servers and an exogenous arrival process remains optimal when the service time distribution is generalized to one with a non-decreasing hazard rate. It remains to be determined whether the same is true of our system. For heterogeneous servers, we proposed the MLRW policy. The second term used in that policy (see (9)) has the total number of customers currently waiting or receiving service as a component but this term uses no information about the elapsed service times of customers currently in service. With non-exponential service time distributions, it might be possible to improve the MLRW policy by incorporating such information. We leave the investigation of these issues for the SQ and MLRW policies for future research.

Appendix A. Proofs

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2011.09.041.

References

Albright, S.C., 1980. Optimal maintenance-repair policies for the machine repair problem. Naval Research Logistics Quarterly 27, 17–27.

Carmichael, D.G., 1986a. Erlang loading models in earthmoving. Civil Engineering Systems 3, 118–124.

Carmichael, D.G., 1986b. Shovel-truck queues: A reconciliation of theory and practice. Construction Management and Economics 4, 161–177.

Ching, W.K., 2001. Machine repairing models for production systems. International Journal of Production Economics 70, 257–266.

Cinlar, E., 1972. Optimal operating policy for the machine repair problem with two service stations. Technical Report No. 266-3, Control Analysis Corp., Palo Alto, CA.

Frostig, E., 1993. Optimal policies for machine repairmen problems. Journal of Applied Probability 30, 703–715.

Goheen, L., 1977. On the optimal operating policy for the machine repair problem when failure and repair times have Erlang distribution. Operations Research 25, 484–492.

Gross, D., Shortle, J.F., Thompson, J.M., Harris, C.M., 2008. Fundamentals of Queueing Theory, fourth ed. Wiley, Hoboken, New Jersey.

Hordijk, A., Koole, G., 1990. On the optimality of the generalized shortest queue policy. Probability in the Engineering and Informational Sciences 4, 477–487.

Hordijk, A., Koole, G., 1992. On the assignment of customers to parallel queues. Probability in the Engineering and Informational Sciences 6, 495–511.

Ke, J., Wang, K., 1999. Cost analysis of the M/M/R machine repair problem with balking, reneging, and server breakdown. Journal of the Operational Research Society 50, 275–282.

Koole, G., 2005. Routing to parallel homogeneous queues. Mathematical Methods in Operations Research, 1–4.

Koole, G., Sparaggis, P.D., Towsley, D., 1999. Minimizing response times and queue lengths in systems of parallel queues. Journal of Applied Probability 36, 1185–1193.

Koyanagi, J., Kawai, H., 1995. An assignment problem for a parallel queueing system with two heterogeneous servers. Mathematical and Computer Modelling 22, 173–181.

Larsen, R.L., Agrawala, A.K., 1983. Control of a heterogeneous two-server exponential queueing system. IEEE Transactions on Software Engineering 9, 522–526.

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

Neely, M.J., Modiano, E., Rohrs, C.E., 2002. Routing over parallel queues with time varying channels with application to satellite and wireless networks. In: Proceedings of Conference on Information Sciences and Systems, Princeton University.

Oil Sands Discovery Centre, 2011. Facts about Albertas oil sands and its industry. http://history.alberta.ca/oilsands/docs/facts_sheets09.pdf, last accessed 25 May 2011.

Righter, R., 1996. Optimal policies for scheduling repairs and allocating heterogenous servers. Journal of Applied Probability 33, 536–547.

Roman, P.A., Daneshmend, L., 2000. Economies of scale in mining-Assessing upper bounds with simulation. The Engineering Economist 45, 326–338.

Stidham, S., Weber, R., 1993. A survey of Markov decision models for control of networks of queues. Qeueueing Systems 13, 291–314.

Ta, C.H., Ingolfsson, A., & Doucette, J. (2010). Haul truck allocation via queueing theory. Working paper, available from <http://apps.business.ualberta.ca/aingolfsson/publications.htm>.

Wartenhorst, P., 1995. N parallel queueing systems with server breakdown and repair. European Journal of Operational Research 82, 302–322.

Weber, R.R., 1978. On the optimal assignment of customers to parallel servers. Journal of Applied Probability 15, 406–413.

Winston, W., 1977. Optimality of the shortest line discipline. Journal of Applied Probability 14, 181–189.

Xu, S.H., Righter, R., Shanthikumar, J.G., 1992. Optimal dynamic assignment of customers to heterogeneous servers in parallel. Operations Research 40, 1126–1138.

Xu, S.H., Zhao, Y.Q., 1996. Dynamic routing and jockeying controls in a two-station queueing system. Advances in Applied Probability 28, 1201–1226.