Probability in the Engineering and Informational Sciences, **28**, 2014, 279–311.
doi:10.1017/S0269964814000011

OPTIMAL STATIC ASSIGNMENT AND ROUTING POLICIES FOR SERVICE CENTERS WITH CORRELATED TRAFFIC

NELSON LEE AND VIDYADHAR G. KULKARNI
Department of Statistics and Operations Research
University of North Carolina
Chapel Hill, NC
USA
E-mail: leent@email.unc.edu; vkulkarn@email.unc.edu

YASUTAKA HIRASAWA
NetApp, Research Triangle Park, NC
USA
E-mail: Yasutaka.Hirasawa@netapp.com

A service center is a facility with multiple heterogeneous servers providing specialized service to multiple types of customers. An assignment policy specifies which server is enabled to serve which types of customer, and a routing policy specifies which server a customer will be routed to for service. Thus, a server can be enabled to serve many types of customer, and a customer may have many alternate servers who can serve him. This paper aims to provide decision models to determine optimal static assignment and routing policies, explicitly taking into account the stochastic fluctuations of demand along with the autocorrelations and cross-correlations of the different traffic streams. We consider several possible performance measures and formulate the optimization problem as a mixed integer nonlinear programming problem. We also develop an efficient heuristic algorithm to enhance scalability. Finally, we compare the different policies using the heuristic algorithms. We observe numerically that the routing policy tries to combine the negatively correlated traffic streams, and separate the positively correlated traffic streams.

1. INTRODUCTION

A service center is a facility with multiple heterogeneous servers providing specialized services to multiple types of customers. An incoming customer of each type requests a service requiring a random amount of time that may depend on customer type and/or server. The arrival processes of the incoming customers of different types form stochastic traffic streams that may be dependent on each other. The service center design and control problems include:

(1) Sizing: determine the number of servers of each type,
(2) Assignment: determine which server to enable to serve which set of customer types,

© Cambridge University Press 2014 0269-9648/14 $25.00

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

(3) Routing: determine which customer should be served by which server,

(4) Scheduling: determine when to activate and de-activate servers.

The objective of this kind of problem is usually to minimize the overall costs while fulfilling the requirements for quality of service, or to maximize appropriate system performance measure given the resource constraints. In this paper, we focus on Problems (2) and (3) above, and provide schemes to determine a static assignment and routing policy that optimizes appropriate system performance measure. Problems (1) and (4) will be treated in a separate paper later on.

Service center design and control problem arises in many practical applications such as cloud computing, data centers, health care facilities, call centers, to name a few. Our study is primarily motivated by the data center design and control problem arising out of our interaction with a technology company. A data center is a facility used to house computers and associated components, such as telecommunications, processing and storage systems. In the past, organizations used to host most of their services on dedicated servers, that is, each server could provide only one service. For example, payroll, inventory management, and sales applications may be hosted on separate servers. A major reason to use dedicated servers is to avoid conflicts between services. However, dedicated servers most likely do not operate at their maximum capacities. They are usually expensive, under-utilized, and energy-consuming. Although no official figures of server utilization in data centers are reported, it has been estimated that the common resource utilization is between 15 and 20% (Vogels [27]). The under-utilized servers result in hardware and energy wastage. Hence, data center design and control with server consolidation is an important topic. Although the motivation of this study is the data center problem, we would like to investigate the general service center design and control problem that is applicable in more general settings.

The service center we consider has multiple servers working in parallel, each with its own queue, and providing specialized services to customers. An incoming customer to the system would request a service of certain type and immediately be routed to one of the servers that is capable of handling this type of service. To benefit from server consolidation, we assume that each server may be capable of handling multiple types of service. In this study, we aim to simultaneously consider two kinds of decisions: the assignment policy and the routing policy, with the objective to optimize a given performance measure, examples of which are introduced in Sections 3–6. The assignment policy determines the set of service types each server is capable of serving. Some examples of assignment policies include: decision of what skills each agent should have in a call center or decision of what software portfolio each computer should have in a data center. On the other hand, the routing policy determines which server a certain type of customer is routed to upon arrival. The formal definitions of the assignment policy and the routing policy are introduced in Section 2. The routing scheme we consider is usually called probabilistic routing or random splitting (Wang and Morris [29]). It is static in the sense that the routing probabilities do not depend on the state of the system, such as the queue lengths. Theoretically, dynamic state-dependent routing policies using queueing information may result in a better system performance. However, this study is motivated by distributed computer systems where gathering such information and implementing dynamic policies accordingly involves a considerable communication overhead and typically nullifies the potential benefits of dynamic policies. Furthermore, our optimal static routing policy will depend upon major system parameters, such as arrival rates, service rates, covariances between streams, etc. In practice, we may monitor these parameters continuously and adjust the optimal routing policy periodically to adapt the changes in these system parameters. We refer readers to Borst [8], Sethuraman and

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

Squillante [23], Guo, Lu, and Squillante [14], and references therein for deeper discussion of the motivation of static routing policy.

1.1. Related work

There is considerable literature on customer routing policies. Borst [8], Buzacott and Shanthikumar [9] (Section 6), and Sethuraman and Squillante [23] provide the structures of the optimal policy and frameworks for determining an optimal routing policy with multiple classes of customers. Shanthikumar and Xu [24] and Guo, Lu, and Squillante [14] also have similar analysis on routing policies but with a single class of customer. All the above mentioned papers mostly focus on optimal routing policies and assume either dedicated or fully flexible servers (i.e., each server can serve any type of customers). Gurvich, Armony, and Mandelbaum [15] and Gurvich and Whitt [16] study the sizing and routing problem of service system with multiple types of customers and servers, but the former paper assumes fully flexible servers and the latter one assumes that the available assignments between type of customer and server are given. Thus, none of these papers consider the assignment and routing policies simultaneously.

Note that these papers assume that each arriving customer must be immediately and permanently routed to one of the feasible servers. It will clearly be better if we can postpone the routing decision until a server becomes free, if such a flexibility was possible. Several researchers have considered such a possibility. For example, Andradóttir, Ayhan, and Down [2,3] and Tekin, Andradóttir, and Down [26] use the fluid model to determine maximum system capacity or throughput under dynamic server assignment policies and provide generalized round-robin policies that achieve the system capacity or throughput arbitrarily close to these upper bounds for queueing networks with flexible servers. However, they do not intend to optimize other system performance measures such as mean waiting time or queue length.

There are many other related papers with specific applications. We discuss some of most related ones in the following three separate subsections based on their applications.

1.1.1. Data center Bichler, Setzer, and Speitkamp [7] and Speitkamp and Bichler [25] provide integer programming models to minimize overall server costs in the data centers. They assume that each type of customer can only be served by one server and consider deterministic capacity demands for each customer type. They formulate the problem as a bin packing problem. Since the bin packing problem is NP-hard (Johnson et al. [19]), they introduce heuristic algorithms to solve the problem. However, these studies do not take into account the system performance requirements.

Chen et al. [10] consider a data center hosting multiple identical servers and providing multiple services. This paper assumes that the servers can be turned on/off with adjustable service rates. The objective is to minimize the operational cost, including electricity cost and setup cost, while satisfying average response time requirements. Anselmi, Amaldi, and Cremonesi [4] and Anselmi, Cremonesi, and Amaldi [5] consider multi-tiered services in the data centers. The objective is to minimize the number of servers used while satisfying performance requirements, such as end-to-end response time constraints and utilization constraints. Utilization constraints, similar to capacity demand constraints in Bichler et al. [7] and Speitkamp and Bichler [25], are linear, while end-to-end response time constraints are nonlinear. They assume that each service tier can only be served by one server, and different application tiers can be served by a common server. They also consider load-balanced system, where traffic can be evenly split to multiple servers. However, these papers do not address the issue of determining the routing policies.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

1.1.2. Call center and contact center Wallace and Whitt [28] propose a staffing algorithm for call centers with performance constraints. Their paper states that the call centers can significantly decrease the number of servers if each agent has two skills instead of one. On the other hand, the additional benefit is not significant when the number of skills for each agent increases beyond two. They provide a heuristic algorithm and use simulation to solve the staffing problems. Their staffing algorithm is similar to our heuristic algorithm for finding the assignment policy, but ours has a different type of routing policy involved.

Whitt [31], Harrison and Zeevi [17], and Bassamboo, Harrison, and Zeevi [6] approach call center problems by using multi-class stochastic fluid models. The objective in these papers is to minimize the sum of staffing cost and expected abandonment cost. Customer abandonment plays a key role in their models. With fluid models, the problems or subproblems are formulated as linear programming problems. However, the same framework may not be suitable for data center problems, because the overflow and customer abandonment are commonly seen in call centers but are less significant in data centers.

1.1.3. Health care system Kwak and Lee [21] present a goal programming model to determine the schedules of physicians and nurses in a health care system. They assume that the demands of physicians and nurses are known and aim to meet both skill and work force requirements, and also to minimize total payroll cost. Jaumard, Semet, and Vovor [18] formulate the nurse staffing and scheduling problem as a mixed integer programming problem. They suppose the nurses have different skill levels and the demand of nurses of each skill level is known. The objective is to minimize the labor costs while satisfying the demand. They further formulate the problem as a shortest path problem to improve the solution in order to satisfy human resource requirements such as workload, off weekends, and rotations.

On the other hand, Green [13], Yankovic and Green [32], and de Véricourt and Jennings [11] provide queueing models to determine appropriate demand levels of resources such as physicians and nurses in a health care system. They mainly formulate the problems as $M/M/s$ and $M/G/s$ queues. The objective is to determine the minimum number of servers needed to satisfy the performance requirements, e.g., the probability of excessive delay.

Many related papers use linear and mixed integer programming models to tackle the problems assuming the demands are known, and use queueing models to determine appropriate server capacities. In this study, we would like to consider the optimization problems taking account the stochastic fluctuations of demand. Moreover, we include correlations of traffic streams in our study. Most literature in this area assumes independent traffic streams. In reality, the traffic streams usually have cycles and are correlated to each other, either positively or negatively. However, very few studies take into account both the stochastic fluctuations and the natural correlation between traffic streams of services. The service center performance can be further improved if we take these factors into consideration (Li [22]).

The rest of the paper is organized as follows. We formulate the problem of finding an optimal assignment and routing policy that minimizes the expected number customers in the system as a mixed integer nonlinear programming problem in Section 2. We describe and analyze a queueing model with multiple-dependent traffic streams in Section 3. Since the analysis of the queueing model is computationally hard, we provide a simple approximation to the objective function in Section 5. Although the computation is now simpler, the problem remains non-convex. Finally, in Section 6 we study an entirely different quadratic objective function that yields a convex mixed integer nonlinear programming problem. This provides a third method of deriving an assignment and routing policy. We then provide a heuristic algorithm to solve these nonlinear mixed integer problems, and use two numerical examples

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

to compare the expected number of customers in the system under the three policies in Section 7. We conclude that the policy out of the third method is the quickest to derive, and does quite well compared to the other two policies. We also observe numerically that the optimal routing policy tries to combine the negatively correlated traffic streams, and separate the positively correlated traffic streams.

2. PROBLEM FORMULATION

Consider a service center having M servers with N specialized service (or customer) types. Each server can provide service to multiple types and a given service type can be handled by multiple servers. Without loss of generality, we assume that an incoming customer requires exactly one type of service. (If they need more than one, we can simply define the combination as a new type.) Let

$$d_{i,k} = \begin{cases} 1, & \text{if server } k \text{ is enabled to provide service type } i, \\ 0, & \text{otherwise,} \end{cases}$$

for $i \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, M\}$. The matrix $d = [d_{i,k}]$ is called the assignment matrix and describes the assignment policy.

We assume that the total arrival process to the system is a Poisson Process (PP) with a fixed rate λ. The inter-class dependence and cross-class dependence of arrival processes are modeled by using a stochastic process $\{Z_n, n \geq 0\}$, where Z_n is the service type of the nth arriving customer. We assume that $\{Z_n, n \geq 0\}$ is an irreducible discrete-time Markov chain (DTMC) with state space $\{1, \ldots, N\}$, transition probability matrix Θ, and steady-state distribution π. We can introduce dependence among the arrival processes of different types of customers by a suitable choice of Θ.

When a customer of type i arrives to the system, it is routed to server k with probability $\alpha_{i,k}$. We assume that the waiting places for customers are with the servers. Hence, an arriving customer needs to be immediately routed to one of the servers that can serve him. A customer of type i can be routed to server k ($\alpha_{i,k} > 0$) only if server k is enabled to provide service type i, that is, $d_{i,k} = 1$. The policy that determines this routing of customers to servers is called a routing policy. The matrix $\alpha = [\alpha_{i,k}]$ is called the routing matrix that describes the static routing policy. The vector $\alpha_k = [\alpha_{1,k}, \ldots, \alpha_{N,k}]'$ is called the routing vector of server k. Each server has a single queue with unlimited space for all classes of customers. The service discipline is first-come-first-served (FCFS) for each server. The service times of customers of type i at server k are iid random variables, with cumulative distribution function (cdf) $F_{i,k}$, mean $\tau_{i,k}$, and variance $\sigma^2_{i,k}$. The service rate of type i customer at server k is $\mu_{i,k} = 1/\tau_{i,k}$. We then define $\tau_k = [\tau_{1,k}, \ldots, \tau_{N,k}]$, $\tau_k^2 = [\tau_{1,k}^2, \ldots, \tau_{N,k}^2]$, and $\sigma_k^2 = [\sigma_{1,k}^2, \ldots, \sigma_{N,k}^2]$.

Now, the arrival rate, the service time distributions, the assignment policy, and the routing policy will determine the performance of the system. Our aim is to identify the static assignment and routing policy that will optimize the system performance. We first introduce the optimization model to determine the optimal assignment and routing policy in the service centers.

First consider a given assignment policy d. A given routing policy α is called *d-feasible* if it only routes a customer to a server that is enabled to serve it, that is

$$d_{i,k} = 0 \Rightarrow \alpha_{i,k} = 0, \text{ for all } i, k.$$

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

For a fixed feasible routing policy, each server can be analyzed as a single-server queue where the inter-arrival times and the service times are modulated by $\alpha_k = [\alpha_{1,k}, \ldots, \alpha_{N,k}]'$ and $\{Z_n, n \geq 0\}$ (see Adan and Kulkarni [1]).

Let $L_k(\alpha_k)$ be the expected number of customers in queue k (including any in service), given a feasible routing policy α. The objective is to minimize the expected total number of customers in the system in steady state. We shall show in the next section that $L_k(\alpha_k)$ is highly nonlinear in α_k.

For a given assignment policy d, we can find an optimal feasible static routing policy by solving the following nonlinear programming problem $P(d)$:

Problem $P(d)$

$$\min \quad \Psi(d, \alpha) = \sum_{k=1}^{M} L_k(\alpha_k), \tag{2.1}$$

$$\text{s.t.} \quad \alpha \text{ is } d\text{-feasible.} \tag{2.2}$$

Let $\alpha^*(d)$ be the optimal d-feasible routing policy obtained by solving $P(d)$. Let

$$\Psi^*(d) = \Psi(d, \alpha^*(d)). \tag{2.3}$$

We next formulate the assignment and routing problem together. First note that for a fixed i and k, any d-feasible policy with $d_{i,k} = 0$ is d-feasible with $d_{i,k} = 1$ (all other components being the same). Hence, $\Psi(d)$ is a decreasing function of each component of d. Thus, in the absence of any further constraints on d, it is optimal set $d_{i,k} = 1$ for all i and k, that is, enable every server to handle each type of customer. In practice, enabling the servers has a cost. There are many ways of modeling such a cost. We handle this in the simplest possible fashion by insisting that at most T of the $d_{i,k}$'s can be set to one, where T is a given integer satisfying $N \leq T \leq NM$. If all assignments cost the same, this is one way of handling the budget constraint. (Alternatively, one can limit the number of assignments on each server or each type of service, or associate costs with setting any $d_{i,k} = 1$ and include a budget constraint.) With this, we can formulate the combined routing and assignment problem as the following mixed integer nonlinear programming program (MINLP):

Problem P

$$\min \quad \Psi^*(d), \tag{2.4}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \sum_{k=1}^{M} d_{i,k} \leq T, \tag{2.5}$$

$$d_{i,k} \in \{0, 1\}, \qquad \forall i \in \{1, \ldots, N\},\ k \in \{1, \ldots, M\}. \tag{2.6}$$

Let d^* be the optimal assignment policy provided by solving P. Then $\alpha^{**} = \alpha^*(d^*)$ is the optimal routing policy. Equation (2.5) guarantees that total number of all assignments does not exceed the limit T.

We need to compute $L_k(\alpha_k)$ in order to solve $P(d)$ and P. We do that in the next section.

3. ANALYSIS OF THE QUEUEING MODEL

Let d be an assignment policy and α be a d-feasible policy. The incoming customer of type i gets routed to queue at server k with probability $\alpha_{i,k}$, and has service time distribution

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

$F_{i,k}(\cdot)$. One can consider the customers being routed to servers other than server k as having zero service times. Let $S_{n,k}$ be the service time of the nth arriving customer (including those with zero service time) to queue k. Thus, customers arrive to queue k according to PP(λ), the type of the nth customer is Z_n, and the service time of a customer of type i is given by

$$G_{i,k}(y) = P(S_{n,k} \le y | Z_n = i) = 1 - \alpha_{i,k}(1 - F_{i,k}(y)).$$

Adan and Kulkarni [1] have analyzed a queueing system of this type. We restate some of their results here.

3.1. Stability

Let $A_i(t)$ be the number of requests of service type i to the system over time $(0,t]$, $Y_{i,k}(t)$ be the number of requests of service type i being routed to server k over time $(0,t]$, and $B_k(t)$ be the number of requests being routing to server k over time $(0,t]$, that is,

$$B_k(t) = \sum_{i=1}^{N} Y_{i,k}(t) = \sum_{i=1}^{N} \text{Bin}(\alpha_{i,k}, A_i(t)).$$

Let $A(t) = \sum_{i=1}^{N} A_i(t)$ be the total number of arrivals over $(0,t]$. Then we know that

$$\lambda = \lim_{t\to\infty} \frac{E(A(t))}{t}.$$

We define the arrival rate of customers of type i as

$$\lambda_i = \lim_{t\to\infty} \frac{E(A_i(t))}{t}.$$

Conditioning on $A(t)$, we obtain

$$\lambda_i = \lim_{t\to\infty} \frac{E(A_i(t))}{t} = \lim_{t\to\infty} \frac{E\left[E\left(\sum_{r=1}^{A(t)} \mathbb{1}_{\{Z_r=i\}} \middle| A(t)\right)\right]}{t} = \lim_{t\to\infty} \frac{E\left(\pi_i A(t)\right)}{t} = \lambda\pi_i, \quad \textbf{(3.1)}$$

and the rate at which customers arrive at queue k is given by

$$\begin{aligned}\lambda_k(\alpha_k) &= \lim_{t\to\infty} \frac{E(B_k(t))}{t} = \lim_{t\to\infty} \frac{E\left(\sum_{i=1}^{N} Y_{i,k}(t)\right)}{t} = \lim_{t\to\infty} \frac{\sum_{i=1}^{N} E[E(\text{Bin}(\alpha_{i,k}, A_i(t))|A_i(t))]}{t} \\ &= \lim_{t\to\infty} \frac{\sum_{i=1}^{N} E(\alpha_{i,k} A_i(t))}{t} = \lambda\pi\alpha_k. \quad \textbf{(3.2)}\end{aligned}$$

The expected service time of a customer arriving at queue k in steady state is given by

$$\tau_k(\alpha_k) = \tau_k \text{diag}[\pi]\alpha_k / \pi\alpha_k. \quad \textbf{(3.3)}$$

Thus, the queue at server k is stable if

$$\lambda_k(\alpha_k)\tau_k(\alpha_k) = \lambda\tau_k \text{diag}[\pi]\alpha_k < 1. \quad \textbf{(3.4)}$$

We shall say the system is stable if queue k is stable for $k = 1, 2, \ldots, M$.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

3.2. The queuing time process

Let $W_{n,k}$ be the waiting time (excluding service time) of the nth customer joining the queue in front of server k. Let

$$\phi_{i,k}^n(s) = E(e^{-sW_{n,k}}; Z_n = i), \quad \text{Re}(s) \geq 0,\ n \geq 0. \tag{3.5}$$

Assume that stability condition holds, and define

$$\phi_{i,k}(s) = \lim_{n\to\infty} \phi_{i,k}^n(s). \tag{3.6}$$

Define the LST of the service time as follows:

$$\tilde{G}_{i,k}(s) = \int_0^\infty e^{-st} dG_{i,k}(t), \tag{3.7}$$

$$\tilde{G}_k(s) = \text{diag}[\tilde{G}_{1,k}(s), \ldots, \tilde{G}_{N,k}(s)]. \tag{3.8}$$

In addition, let e_N be an N-vector whose elements are all one. The main result is given in the following theorem.

THEOREM 1: *(*Adan and Kulkarni [1]*) The transform vector* $\phi_k(s) = [\phi_{1,k}(s), \ldots, \phi_{N,k}(s)]$ *satisfies*

$$\phi_k(s)[\lambda\tilde{G}_k(s)\Theta + (s-\lambda)I_N] = sv_k, \tag{3.9}$$

$$\phi_k(0)e_N = 1, \tag{3.10}$$

where I_N *is an* N*-dimensional identity matrix. Let* Γ_1^k *and* Γ_2^k *be the first and second moments of service times at server* k*:*

$$\Gamma_1^k = diag[\alpha_{1,k}\tau_{1,k}, \ldots, \alpha_{N,k}\tau_{N,k}], \tag{3.11}$$

$$\Gamma_2^k = diag[\alpha_{1,k}(\sigma_{1,k}^2 + \tau_{1,k}^2), \ldots, \alpha_{N,k}(\sigma_{N,k}^2 + \tau_{N,k}^2)]. \tag{3.12}$$

The vector $v_k = [v_{1,k}, \ldots, v_{N,k}]$ *is given by the unique solution to the following* N *linear equations:*

$$v_k a_i = 0, \quad i \in \{2, \ldots, N\}, \tag{3.13}$$

$$v_k\lambda^{-1}e_N = \pi(\lambda^{-1}I_N - \Gamma_1^k)e_N, \tag{3.14}$$

where a_i *is a non-zero vector satisfying*

$$[\lambda\tilde{G}_k(s_i)\Theta + (s_i - \lambda)I_N]a_i = 0, \quad i \in \{2, \ldots, N\}, \tag{3.15}$$

and s_i *is the solution of* s *to*

$$\det(\lambda\tilde{G}_k(s)\Theta + (s-\lambda)I_N) = 0, \tag{3.16}$$

with $s_1 = 0$ *and* $Re(s_i) > 0$ *for* $i = 2, \ldots, N$.

The solution of Eq. (3.16) involves a nonlinear eigenvalue problem, which is computationally difficult to solve when the dimension is high.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

3.3. Expected waiting time

Define

$$m_{i,k} = \lim_{n\to\infty} E(W_{n,k}; Z_n = i), \tag{3.17}$$

$$m_k = [m_{1,k}, \ldots, m_{N,k}]. \tag{3.18}$$

THEOREM 2: (Adan and Kulkarni [1]) *The vector m_k satisfies the following equations:*

$$m_k(I_N - \Theta) = \pi(\Gamma_1^k\Theta - \lambda^{-1}I_N) + v_k\lambda^{-1}I_N, \tag{3.19}$$

$$m_k(\lambda^{-1}I_N - \Gamma_1^k)e_N = \frac{1}{2}\pi\Gamma_2^k e_N, \tag{3.20}$$

where v_k is as in Theorem 1.

The expected queueing time in queue of server k is then given by $m_k e_N$. Thus, the expected queueing time in queue plus service time of customers being routed to server k is

$$\left(m_k + \frac{\pi\Gamma_1^k}{\pi\alpha_k}\right)e_N. \tag{3.21}$$

3.4. Expected queue length

By Little's law and the result from Eq. (3.21), we know that the expected queue length of server k is given by

$$L_k(\alpha_k) = \lambda\pi\alpha_k\left(m_k + \frac{\pi\Gamma_1^k}{\pi\alpha_k}\right)e_N. \tag{3.22}$$

This is a highly nonlinear function of α_k, and difficult to compute due to necessity of solving Eq. (3.16).

4. OPTIMAL ASSIGNMENT AND ROUTING POLICIES

With the results of the previous section, we can model the routing problem for a given assignment policy d as a nonlinear programming problem as follow:

Problem $P(d)$

$$\min \quad \Psi(d,\alpha) = \sum_{k=1}^{M}\left(m_k + \frac{\pi\Gamma_1^k}{\pi\alpha_k}\right)e_N, \tag{4.1}$$

$$\text{s.t.} \quad \sum_{k=1}^{M}\alpha_{i,k} = 1, \qquad \forall i \in \{1,\ldots,N\}, \tag{4.2}$$

$$\alpha_{i,k} \le d_{i,k}, \qquad \forall i \in \{1,\ldots,N\},\ k \in \{1,\ldots,M\}, \tag{4.3}$$

$$\lambda\tau_k\text{diag}[\pi]\alpha_k < 1, \qquad \forall k \in \{1,\ldots,M\}, \tag{4.4}$$

$$\alpha_{i,k} \ge 0, \qquad \forall i \in \{1,\ldots,N\},\ k \in \{1,\ldots,M\}. \tag{4.5}$$

Equation (4.2) guarantees that the traffic of each type is routed to at least one server, while Eq. (4.3) prevents the traffic of any class from being routed to a server that is not enabled to handle it (i.e., the routing policy is d-feasible). Equation (4.4) is the stability constraint.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

Let $\alpha^*(d)$ be the optimal routing policy provided by solving $P(d)$. Define

$$\Psi^*(d) = \Psi(d, \alpha^*(d)) = \sum_{k=1}^{M} L_k(\alpha_k^*(d)). \tag{4.6}$$

We can model the combined assignment and routing problem as:

Problem P

$$\min \quad \Psi^*(d), \tag{4.7}$$

$$\text{s.t.} \quad \sum_{i=1}^{N}\sum_{k=1}^{M} d_{i,k} \le T, \tag{4.8}$$

$$d_{i,k} \in \{0,1\}, \qquad \forall i \in \{1,\dots,N\},\ k \in \{1,\dots,M\}. \tag{4.9}$$

Let d^* be the optimal assignment policy obtained by solving the above nonlinear mixed integer problem P. Then $\alpha^{**} = \alpha^*(d^*)$ is the optimal routing policy. Note that the objective function Eq. (4.1) is not in a closed form since the m_k's are obtained by the matrix analytic method as described in Sections 3.2 and 3.3. This queueing model provides the exact expected number in the system. However, the calculation is complicated and makes $L_k(\alpha_k)$ difficult to be used in the objective function. Hence, we develop an approximation for L_k in the next section.

5. DIFFUSION APPROXIMATION

In this section, we introduce a diffusion approximation to estimate the expected queue lengths when traffic intensity is high. We define $\tilde{L}_k(\alpha_k)$ as an approximation to $L_k(\alpha_k)$.

Define the long-run variance–covariance matrix $\Sigma = [\Sigma_{i,j}]$ as

$$\Sigma_{i,j} = \lim_{t\to\infty} \frac{\mathrm{Cov}(A_i(t), A_j(t))}{t}, \quad i,j \in \{1,\dots,N\}. \tag{5.1}$$

The next theorem shows how to compute this Σ.

THEOREM 3: *Suppose the arrival process is modulated by a DTMC* $\{Z_n, n \ge 0\}$ *as described in Section 2. Then the variance–covariance matrix* Σ *is given by*

$$\begin{aligned}\Sigma = \lambda\{&\mathrm{diag}[\pi] + \mathrm{diag}[\pi][(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}] \\ &+ [(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}]'\,\mathrm{diag}[\pi]\}.\end{aligned} \tag{5.2}$$

See Appendix 8 for a detailed proof.

Similar to the queuing model discussed in Section 3, we consider the limiting behavior of one single server at a time. For any server k, the inter-arrival times and service times are regulated by α_k and $\{Z_n, n \ge 0\}$. However, unlike the analysis of queueing model, we only consider the customers that are actually routed to each server. For any given k, we define $\{U_{n,k}, n \ge 1\}$ to be the sequence of inter-arrival times to server k. Clearly, this is not an iid sequence, and hence the arrival process $\{B_k(t), t \ge 0\}$ generated by it is not a renewal process. Similarly, let $\{V_{n,k}, n \ge 1\}$ be the sequence of service times at server k. It is also not an iid sequence, and hence the queue at server k is not a $GI/GI/1$ queue.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

Our first step is to approximate it by a $GI/GI/1$ queue. To do this, we construct an iid sequence $\{\tilde{U}_{n,k}, n \geq 1\}$ of inter-arrival times, so that the first two moments of the arrival process $\{\tilde{B}_k(t), t \geq 0\}$ generated by it match the first two moments of $\{B_k(t), t \geq 0\}$. The precise statement is given in the theorem below.

THEOREM 4: *Let $\{\tilde{U}_{n,k}, n \geq 1\}$ be an iid sequence of non-negative random variables and $\{\tilde{B}_k(t), t \geq 0\}$ be the renewal process generated by it. Suppose*

$$E(\tilde{U}_{n,k}) = \frac{1}{\lambda \pi \alpha_k}, \tag{5.3}$$

$$\mathrm{Var}(\tilde{U}_{n,k}) = \frac{\alpha_k'(\Sigma - \lambda \,\mathrm{diag}[\pi])\alpha_k + \lambda \pi \alpha_k}{(\lambda \pi \alpha_k)^3}. \tag{5.4}$$

Then

$$\lim_{t\to\infty} \frac{E(\tilde{B}_k(t))}{t} = \lim_{t\to\infty} \frac{E(B_k(t))}{t}, \tag{5.5}$$

$$\lim_{t\to\infty} \frac{\mathrm{Var}(\tilde{B}_k(t))}{t} = \lim_{t\to\infty} \frac{\mathrm{Var}(B_k(t))}{t}. \tag{5.6}$$

The detailed proof is given in Appendix A.2.

We also construct an iid sequence $\{\tilde{V}_{n,k}, n \geq 1\}$ of service times whose first two moments match the first two moments of $\{V_{n,k}, n \geq 1\}$. The precise statement is given in the theorem below.

THEOREM 5: *Let $\{\tilde{V}_{n,k}, n \geq 1\}$ be an iid sequence of non-negative random variables with*

$$E(\tilde{V}_{n,k}) = \frac{\tau_k \,\mathrm{diag}[\pi]\alpha_k}{\pi \alpha_k}, \tag{5.7}$$

$$\mathrm{Var}(\tilde{V}_{n,k}) = \frac{(\sigma_k^2 + \tau_k^2)\,\mathrm{diag}[\pi]\alpha_k}{\pi \alpha_k} - \left(\frac{\tau_k \,\mathrm{diag}[\pi]\alpha_k}{\pi \alpha_k}\right)^2. \tag{5.8}$$

Then

$$E(\tilde{V}_{n,k}) = \lim_{n\to\infty} E(V_{n,k}), \tag{5.9}$$

$$\mathrm{Var}(\tilde{V}_{n,k}) = \lim_{n\to\infty} \mathrm{Var}(V_{n,k}). \tag{5.10}$$

The detailed proof is given in Appendix A.3.

Now we consider a $GI/GI/1$ queue with arrival process $\{\tilde{B}_k(t), t \geq 0\}$ and service times $\{\tilde{V}_{n,k}, n \geq 1\}$. From Theorems 4 and 5 we can further write down the traffic intensity, ρ_k, squared coefficient of variation of the inter-arrival times, $c_{a_k}^2$, and squared coefficient of variation of the service times, $c_{s_k}^2$, as:

$$\rho_k = \lambda \tau_k \,\mathrm{diag}[\pi]\alpha_k, \tag{5.11}$$

$$c_{a_k}^2 = \frac{\mathrm{Var}(\tilde{U}_{n,k})}{[E(\tilde{U}_{n,k})]^2} = \frac{\alpha_k'(\Sigma - \lambda \,\mathrm{diag}[\pi])\alpha_k + \lambda \pi \alpha_k}{\lambda \pi \alpha_k}, \tag{5.12}$$

$$c_{s_k}^2 = \frac{\mathrm{Var}(\tilde{V}_{n,k})}{[E(\tilde{V}_{n,k})]^2} = \frac{(\sigma_k^2 + \tau_k^2)\,\mathrm{diag}[\pi]\alpha_k \pi \alpha_k - (\tau_k \,\mathrm{diag}[\pi]\alpha_k)^2}{(\tau_k \,\mathrm{diag}[\pi]\alpha_k)^2}. \tag{5.13}$$

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

We then use the diffusion approximation from Whitt [30] for the expected queue length of server k:

$$\begin{aligned}\tilde{L}_k(\alpha_k) &= \frac{\rho_k c_{a_k}^2 + c_{s_k}^2}{2(1-\rho_k)} \\ &= \frac{(\alpha_k'(\Sigma - \lambda \operatorname{diag}[\pi])\alpha_k + \lambda\pi\alpha_k - 1)(\tau_k \operatorname{diag}[\pi]\alpha_k)^2 + (\sigma_k^2 + \tau_k^2)\operatorname{diag}[\pi]\alpha_k\pi\alpha_k}{2(1-\lambda\tau_k \operatorname{diag}[\pi]\alpha_k)(\tau_k \operatorname{diag}[\pi]\alpha_k)^2}.\end{aligned} \tag{5.14}$$

Whitt [30] shows that $\tilde{L}_k(\alpha_k)$ is a good approximation for the expected queue length of $GI/GI/1$ queue, especially in heavy traffic. We use numerical examples in Section 7.2 to show that this approximation works well for our study. Guo, Lu, and Squillante [14] also derive a similar diffusion approximation for the expected queue length and use it to obtain optimal routing policy of single class customers to multiple servers.

Using Eq. (5.14) as performance measure, we model the routing problem for a given assignment policy d as a nonlinear programming problem as follow:

Problem $\tilde{P}(d)$

$$\min \tilde{\Psi}(d,\alpha) = \sum_{k=1}^{M} \frac{\begin{array}{r}(\alpha_k'(\Sigma - \lambda \operatorname{diag}[\pi])\alpha_k + \lambda\pi\alpha_k - 1)(\tau_k \operatorname{diag}[\pi]\alpha_k)^2 \\ + (\sigma_k^2 + \tau_k^2)\operatorname{diag}[\pi]\alpha_k\pi\alpha_k\end{array}}{2(1-\lambda\tau_k \operatorname{diag}[\pi]\alpha_k)(\tau_k \operatorname{diag}[\pi]\alpha_k)^2}, \tag{5.15}$$

$$\text{s.t.} \quad \sum_{k=1}^{M} \alpha_{i,k} = 1, \qquad \forall i \in \{1,\ldots,N\}, \tag{5.16}$$

$$\alpha_{i,k} \le d_{i,k}, \qquad \forall i \in \{1,\ldots,N\},\ k \in \{1,\ldots,M\}, \tag{5.17}$$

$$\lambda\tau_k \operatorname{diag}[\pi]\alpha_k < 1, \qquad \forall k \in \{1,\ldots,M\}, \tag{5.18}$$

$$\alpha_{i,k} \ge 0, \qquad \forall i \in \{1,\ldots,N\},\ k \in \{1,\ldots,M\}. \tag{5.19}$$

As in the queueing model, let $\tilde{\alpha}^*(d)$ be the optimal routing policy obtained by solving $\tilde{P}(d)$. Define

$$\tilde{\Psi}^*(d) = \tilde{\Psi}(d, \tilde{\alpha}^*(d)) = \sum_{k=1}^{M} \tilde{L}_k(\alpha_k^*(d)). \tag{5.20}$$

Then we formulate the combined assignment and routing problem as:

Problem $\tilde{P}$

$$\min \quad \tilde{\Psi}^*(d), \tag{5.21}$$

$$\text{s.t.} \quad \sum_{i=1}^{N}\sum_{k=1}^{M} d_{i,k} \le T, \tag{5.22}$$

$$d_{i,k} \in \{0,1\}, \qquad \forall i \in \{1,\ldots,N\},\ k \in \{1,\ldots,M\}. \tag{5.23}$$

Let $\tilde{d}^*$ be the optimal assignment policy obtained by solving the above nonlinear mixed integer problem $\tilde{P}$. Then $\tilde{\alpha}^{**} = \tilde{\alpha}^*(d^*)$ is the optimal routing policy. One advantage of this approximation is that the parameters for this model are easier to estimate. To solve the problem P, introduced in the Section 4, we need to obtain the transition probability matrix.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

However, to accurately estimate the transition probability matrix, we have to observe the sequences of incoming traffic, which could be difficult due to possible multiple arrivals with the same time stamp. On the other hand, to solve the problem $\tilde{P}$, we only need to observe the total incoming traffic in a given time interval for each class to estimate the steady-state mean arrival rates and the variance–covariance matrix. Then we can approximate the expected queue length using this limited information of traffic. Another advantage of the approximation is that the approximated expected queue length $\tilde{L}_k(\alpha_k)$ can be obtained as a closed-form expression. Compared to using a matrix analytic method to obtain the expected queue length $L_k(\alpha_k)$ in queueing model, using the closed form expression involving only matrix multiplication in diffusion approximation model is obviously preferable and much faster.

However, neither $L_k(\alpha_k)$ nor $\tilde{L}_k(\alpha_k)$ are convex functions of α_k in general, even if we further assume that the service time distributions are the same for all classes of customers. For example, suppose that the service time is exponentially distributed with rate μ for all types of customers on every server. The Hessian matrix of $\tilde{L}_k(\alpha_k)$ with respect to α_k is

$$\tilde{H}_k(\alpha_k) = \frac{[\alpha_k\lambda\pi + (\mu - \lambda\pi\alpha_k)I_N]'(\Sigma - \lambda\,\mathrm{diag}[\pi])[\alpha_k\lambda\pi + (\mu - \lambda\pi\alpha_k)I_N]}{(\mu - \lambda\pi\alpha_k)^3} + \frac{2\mu\lambda^2\pi'\pi}{(\mu - \lambda\pi\alpha_k)^3}. \tag{5.24}$$

Unfortunately, $\tilde{H}_k(\alpha_k)$ is not a positive-semidefinite matrix because $(\Sigma - \lambda\,\mathrm{diag}[\pi])$ may not be positive-semidefinite. This means the objective functions of P and $\tilde{P}$ may not be convex. The following theorem provides an analytical result for a special case.

THEOREM 6: *If traffic streams of all services are independent of each other and can be routed to any server, that is, $d_{i,k} = 1$ for all i and k, and the service time distributions are the same for all services, then*

$$\alpha_{i,k} = \frac{1}{M}, \quad \forall i \in \{1, \ldots, N\},\ k \in \{1, \ldots, M\} \tag{5.25}$$

are optimal solutions to P and $\tilde{P}$.

PROOF: Under any static routing policy, if arrival processes of all services are independent, then the arrival process routed to each server is a PP. Since the service time distributions are all the same, the queueing system at each server forms an $M/G/1$ queue. We will show that the sum of expected queue lengths over all servers is minimized when the traffic intensity is the same for every server. If all assignments are available on every server, then $\alpha_{i,k} = 1/M$ for all i and k is the solution that always makes traffic intensity the same for every server and thus is optimal. The detailed proof is in Appendix A.4. ■

If traffic streams of all services are independent of each other and can be routed to any server, but service time distributions are not all identical, Borst [8] has shown that $L_k(\alpha_k)$ is convex in α and provided a framework for solving the routing problem.

Remark 1: One can show by counter example that Eq. (5.25) is not an optimal solution to P or $\tilde{P}$ if traffic streams of some services are correlated with each other, that is, there are some non-zero off-diagonal entries in Σ.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

6. CONVEX QUADRATIC MODEL

The performance measure introduced in the queueing model of Section 3 is difficult to compute. Hence, we introduced diffusion approximation model in Section 5. Although this produces analytically tractable performance measure, the resulting optimization problem remains non-convex, which makes its solution hard. This motivates us to find an alternate performance measure that can efficiently provide a good assignment and routing policy (which might not be optimal). Note that this new performance measure is not meant to be a further approximation to the performance measures of Section 3 or 5. Its main utility is in producing a candidate assignment and routing policy. Then we can compare performance of this policy with those obtained in Sections 3 and 5 using the performance measure of Section 3. In this section, we derive a convex quadratic model that is applicable when the service time distributions only depend on the servers but not on the customer types, that is, $F_{i,k} = F_k \ \forall i, k$ and F_k is a non-negative random variable. In this case, we know $\mu_{i,k} = \mu_k = 1/E(F_k) \ \forall i, k$, and may interpret μ_k as the service rate or service capacity of server k. Let $A_i(t)$, $B_k(t)$, λ_i, and $\Sigma_{i,j}$ be as defined in Section 3, Eqs. (3.1) and (5.1). We further assume that the system starts in the steady state at time $t = 0$. Then the total traffic being routed to server k in a unit of time is

$$B_k(1) = \sum_{i=1}^{N} \text{Bin}(\alpha_{i,k}, A_i(1)).$$

The service capacity of server k is μ_k as described earlier. Hence, we can think of $\mu_k - B_k(1)$ as the capacity imbalance in a unit of time. Now we define the performance measure $Q_k(\alpha_k)$ as given below:

$$\begin{aligned} Q_k(\alpha_k) &= E\left[\mu_k - B_k(1)\right]^2 = \text{Var}[\mu_k - B_k(1)] + \{E[\mu_k - B_k(1)]\}^2 \\ &= \text{Var}[B_k(1)] + \{E[\mu_k - B_k(1)]\}^2 \\ &= \alpha_k'(\hat{\Sigma} - \lambda\,\text{diag}[\pi])\alpha_k + \lambda\pi\alpha_k + (\lambda\pi\alpha_k - \mu_k)^2 \\ &= \alpha_k'\hat{\Sigma}\alpha_k + (\lambda\pi\alpha_k - \mu_k)^2 - \alpha_k'(\lambda\,\text{diag}[\pi])\alpha_k + \lambda\pi\alpha_k, \end{aligned} \tag{6.1}$$

where $\hat{\Sigma} = [\hat{\Sigma}_{ij}]$, with

$$\hat{\Sigma}_{ij} = \text{Cov}(A_i(1), A_j(1)).$$

Note that $\hat{\Sigma}$ can be approximated by Σ if the number of arrivals in one unit of time is large; see Eq. (A.1). The derivation of the above equation is similar to the proof of Eq. (A.9) for Theorem 4. It can be shown that $Q_k(\alpha_k)$ is not a convex function since $(\hat{\Sigma} - \lambda\,\text{diag}[\pi])$ may not be positive-semidefinite. To avoid this non-convex issue, we define another version of capacity imbalance in a unit of time as $\mu_k - \hat{B}_k(1)$, where

$$\hat{B}_k(1) = \sum_{i=1}^{N} E[\text{Bin}(\alpha_{i,k}, A_i(1))|A_i(1)] = \sum_{i=1}^{N} \alpha_{i,k} A_i(1),$$

the conditional expected total traffic being routed to server k in a unit of time given the number of arrivals of each type to the system. Similar to Eq. (6.1), we define the performance measure $\hat{Q}_k(\alpha_k)$ as follows:

$$\hat{Q}_k(\alpha_k) = E\left[\mu_k - \hat{B}_k(1)\right]^2. \tag{6.2}$$

Let

$$\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_N].$$

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

THEOREM 7: *We have*

$$\hat{Q}_k(\alpha_k) = \alpha_k' \hat{\Sigma} \alpha_k + (\Lambda \alpha_k - \mu_k)^2. \tag{6.3}$$

$\hat{Q}_k(\alpha_k)$ *is a convex quadratic function of* α_k.

PROOF: By definition, we first derive $\hat{Q}_k(\alpha_k)$ as

$$\begin{aligned}\hat{Q}_k(\alpha_k) &= E\left[\mu_k - \hat{B}_k(1)\right]^2 = \mathrm{Var}[\mu_k - \hat{B}_k(1)] + \{E[\mu_k - B_k(1)]\}^2 \\ &= \mathrm{Var}[\hat{B}_k(1)] + \{E[\mu_k - \hat{B}_k(1)]\}^2 \\ &= \mathrm{Var}\left[\sum_{i=1}^{N} A_i(1)\alpha_{i,k}\right] + \left\{E\left[\mu_k - \sum_{i=1}^{N} A_i(1)\alpha_{i,k}\right]\right\}^2 = \alpha_k' \hat{\Sigma} \alpha_k + (\Lambda \alpha_k - \mu_k)^2.\end{aligned} \tag{6.4}$$

Next, we show the convexity. The Jacobian matrix of $\hat{Q}_k(\alpha_k)$ with respect to α_k is

$$J_{\hat{Q}_k}(\alpha_k) = \hat{\Sigma}\alpha_k + 2\Lambda'(\Lambda\alpha_k - \mu_k), \tag{6.5}$$

and the Hessian matrix of $\hat{Q}_k(\alpha_k)$ with respect to α_k is

$$H_{\hat{Q}_k}(\alpha_k) = \hat{\Sigma} + 2\Lambda'\Lambda. \tag{6.6}$$

This Hessian matrix is positive-semidefinite because $\hat{\Sigma}$ is a variance–covariance matrix, which is always positive-semidefinite, and $2\Lambda'\Lambda$ is positive-semidefinite as well. ■

There are two main advantages of this objective function $\hat{Q}_k(\alpha_k)$. First, we need to estimate only the first and second moments of traffic streams to evaluate it. Thus, it is easier to use than the objective function $L_k(\alpha_k)$. Also, it uses as much information as the objective function $\tilde{L}_k(\alpha_k)$. However, unlike $\tilde{L}_k(\alpha_k)$ or $Q_k(\alpha_k)$, $\hat{Q}_k(\alpha_k)$ is convex. Note that we are not claiming that $\hat{Q}_k(\alpha_k)$ is an approximation of $L_k(\alpha_k)$ or $\tilde{L}_k(\alpha_k)$. The main motivation is to provide a convex quadratic program to obtain a candidate assignment and routing policy. Using $\hat{Q}_k(\alpha_k)$ as performance measure, we model the routing problem for a given assignment policy d as a convex quadratic programming problem as follows:
Problem $P^q(d)$

$$\min \quad \Psi^q(d,\alpha) = \sum_{k=1}^{M} [\alpha_k' \hat{\Sigma} \alpha_k + (\Lambda\alpha_k - \mu_k)^2], \tag{6.7}$$

$$\text{s.t.} \quad \sum_{k=1}^{M} \alpha_{i,k} = 1, \qquad \forall i \in \{1,\ldots,N\}, \tag{6.8}$$

$$\alpha_{i,k} \le d_{i,k}, \qquad \forall i \in \{1,\ldots,N\},\ k \in \{1,\ldots,M\}, \tag{6.9}$$

$$\Lambda\alpha_k < \mu_k, \qquad \forall k \in \{1,\ldots,M\}, \tag{6.10}$$

$$\alpha_{i,k} \ge 0, \qquad \forall i \in \{1,\ldots,N\},\ k \in \{1,\ldots,M\}. \tag{6.11}$$

Let $\alpha^q(d)$ be the optimal routing policy provided by solving $P^q(d)$. Define

$$\Psi^q(d) = \Psi^q(d, \alpha^q(d)) = \sum_{k=1}^{M} \hat{Q}_k(\alpha_k^q(d)). \tag{6.12}$$

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

As in the previous two models, we formulate the combined assignment and routing problem as:

Problem P^q

$$\min \quad \Psi^q(d), \tag{6.13}$$

$$\text{s.t.} \quad \sum_{i=1}^{N}\sum_{k=1}^{M} d_{i,k} \le T, \tag{6.14}$$

$$d_{i,k} \in \{0,1\}, \qquad \forall i \in \{1,\ldots,N\},\ k \in \{1,\ldots,M\}. \tag{6.15}$$

P^q can be solved as a mixed integer quadratic programming problem (MIQP). There are efficient solvers available to solve this type of problems, such as CPLEX and Gurobi. Let d^q be the optimal assignment policy obtained by solving the above MIQP problem P^q. Then $\alpha^{q*} = \alpha^q(d^q)$ is the optimal routing policy.

Remark 2: One can show that the variance of $\hat{B}_k(1)$,

$$\text{Var}[\hat{B}_k(1)] = \alpha_k' \hat{\Sigma} \alpha_k.$$

The folk theorem in queues says that the congestion can be reduced by reducing the variance of the input process. Thus, it would make sense to simply minimize

$$\sum_{k=1}^{M} \alpha_k' \hat{\Sigma} \alpha_k.$$

We have numerically evaluated this objective function and found that it performs much worse than $\sum_{k=1}^{M} \hat{Q}_k$. It produces policies that substantially under-perform the policies produced. Thus, somehow the term $(\Lambda\alpha_k - \mu_k)^2$ plays a very discriminating part in this problem.

Similar to queueing model and diffusion approximation model, we next consider a further special case where all assignments are available on every server. In this case, we have the following analytical solution to P^q. Let L be a subset of {1,. . . ,M} so that

$$1 + \lambda + |L|\mu_k - \sum_{l \in L} \mu_l \ge 0, \qquad \forall k \in L, \tag{6.16}$$

$$1 + \lambda + |L|\mu_k - \sum_{l \in L} \mu_l < 0, \qquad \forall k \notin L. \tag{6.17}$$

THEOREM 8: *Let* $d_{i,k} = 1$ *for all* i *and* k, *and* L *be as defined above. Then the optimal routing policy for* P^q *is given by*

$$\alpha_{i,k} = \begin{cases} \dfrac{1 + \lambda + |L|\mu_k - \sum_{l \in L} \mu_l}{(1+\lambda)|L|}, & \forall i \in \{1,\ldots,N\},\ k \in L, \\ 0, & \text{otherwise.} \end{cases} \tag{6.18}$$

The optimality of Eq. (6.18) can be shown by verifying that the Karush–Kuhn–Tucker conditions are satisfied by this solution. See Appendix A.5. for the detailed proof.

Remark 3: We describe an O(M) algorithm to find L defined above. Without loss of generality, we assume that the servers are listed in the descending order of service rate, that is, $\mu_1 \ge \cdots \ge \mu_M$.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

Algorithm 1

Finding the Set L

$L \leftarrow \{1\}$;
for $k = 2 \rightarrow M$ **do**
  if $1 + \lambda + |L|\mu_k - \sum_{l \in L} \mu_l \geq 0$ **then**
    $L \leftarrow L \cup \{k\}$;
  else
    break;
  end if
end for

This procedure can be justified by discussing the two possible outcomes of the "if" statement for any k within the loop:

Case I: When the condition of "if" statement is satisfied,

$$1 + \lambda + (|L| + 1)\mu_j - \sum_{l \in L \cup \{k\}} \mu_l \geq 1 + \lambda + |L|\mu_k - \sum_{l \in L} \mu_l \geq 0, \quad \forall j \in \{k\} \cup L = \{1, \ldots, k-1\}. \tag{6.19}$$

In other words, adding new element into the set L will not nullify the existing elements in L. Hence, we update the set L to $L \cup \{k\}$ and loop to the next.

Case II: When the condition of "if" statement is not satisfied,

$$1 + \lambda + (|L| + 1)\mu_j - \sum_{l \in L \cup \{k\}} \mu_l \leq 1 + \lambda + |L|\mu_k - \sum_{l \in L} \mu_l < 0, \quad \forall j \notin L = \{1, \ldots, k-1\}. \tag{6.20}$$

In other words, adding any other elements into the set L will not satisfy the condition. Hence, we stop as soon as we obtain the first violation of the condition and the set L has been determined.

7. SOLUTION

There are commercial software packages available to solve our convex quadratic model P^q. For example, the well known AMPL/CPLEX is capable of solving the MIQP problem. On the other hand, for queueing model P or diffusion approximation model $\tilde{P}$, the global optimal solutions are difficult to obtain because the objective functions are non-convex and some decision variables are binary. We have discussed some special cases that can be solved analytically in previous sections. Beyond those special cases, we need to use heuristic algorithms to solve the problem in general. We introduce one such algorithm below.

7.1. Heuristic Algorithm

The main goal of the heuristic algorithm is to determine the optimal static routing and assignment policy. The assignment policy is obtained by solving the nonlinear integer programs P or $\tilde{P}$, which are difficult to solve in general. Meanwhile, determining the optimal

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

static routing scheme for a fixed assignment policy involves solving $P(d)$ or $\tilde{P}(d)$, which are relatively easy since they involve a continuous nonlinear optimization.

We introduce a heuristic algorithm called Backward Selection Heuristic Algorithm. In this algorithm, we start by assuming all assignments are available on every server, that is, $d_{i,k} = 1$ for all i and k, and finding the optimal static routing policy, α, under this assumption.

We have shown in Theorem 6 that $\alpha_{i,k}$ is positive for all i and k if traffic streams are independent. However, if traffic streams are not independent, we can expect to have $\alpha_{i,k}$ equal to zero for some i and k. The intuitive explanation is that the system would benefit from routing the traffic streams with negative correlations into common servers but suffer from routing the traffic streams with positive correlations into common ones. Hence, if traffic streams of service type i and j are positively correlated, usually $\alpha_{i,k}$ and $\alpha_{j,k}$ would not be positive at the same time. We do not have a rigorous proof of this, but we will illustrate this idea by a numerical example in Section 7.2.

Based on this initial routing policy, the second step is to remove all assignments with optimal solution $\alpha_{\cdot,\cdot} = 0$, that is, if $\alpha_{i,k} = 0$ then set $d_{i,k} = 0$. In this step, we remove the unused assignments so that we can decrease the total number of assignments without sacrificing system performance. Then we check whether total number of assignments left is less than or equal to the desired number T. If yes, the solution satisfies all constraints of the mixed integer nonlinear programming problem and is optimal. Otherwise, further elimination of assignments is needed.

In the next step, we remove an assignment on a server that results in the smallest increase in the objective function value. The idea of this algorithm is to behave in a greedy fashion. We try to remove one "least important" assignment at a time until total number of assignments left is no more than T. It may not result in an optimal solution but can provide a good solution in a relatively short time. It is common in the service center design problem that practitioners pursue a good solution instead of an optimal solution since finding optimal solution requires too much effort. Also, the greatly fluctuating traffic streams in service center makes an accurate design unnecessary. To find the solution, this algorithm has to run $O(M^2N^2)$ nonlinear programming problems in the worst case. It takes a long time when M and N are large, but we can expect it takes much shorter time than solving the original problem. The pseudocode of this algorithm is presented in Appendix B.

In the next subsection, we use a numerical example to illustrate the Backward Selection Heuristic Algorithm by applying queueing model and diffusion approximation model as congestion performance measures. We will also compare these two models with convex quadratic model at the end.

Note that we do not use this algorithm to solve P^q, since it is a mixed integer quadratic program and there are standard software packages available to solve it.

7.2. Numerical Example I

We consider an example with five servers ($M = 5$) and eight types of services ($N = 8$). We assume that the overall arrival rate, the transition probability matrix of the DTMC determining the customer class, and the service time distributions are known. We can calculate the variance–covariance of arrival process needed for diffusion approximation model from the given arrival rate and transition probability matrix. The upper limit of the total number of assignments is $T = 12$.

We assume that the overall customer arrival process is a PP with rate $\lambda = 135$ and the service time is exponentially distributed with rate dependent on server. The service rates

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

of five servers are 5, 5, 20, 40, and 80, respectively. Let transition probability matrix

$$\Theta = \begin{bmatrix} 0.0333 & 0.3000 & 0.1905 & 0.0952 & 0.0476 & 0.0994 & 0.1068 & 0.1271 \\ 0.2667 & 0.0667 & 0.1905 & 0.0952 & 0.0476 & 0.0994 & 0.1068 & 0.1271 \\ 0.1569 & 0.1765 & 0.3000 & 0.0167 & 0.0167 & 0.0994 & 0.1068 & 0.1271 \\ 0.1569 & 0.1765 & 0.0333 & 0.2833 & 0.0167 & 0.0994 & 0.1068 & 0.1271 \\ 0.1569 & 0.1765 & 0.0667 & 0.0333 & 0.2333 & 0.0994 & 0.1068 & 0.1271 \\ 0.1569 & 0.1765 & 0.1905 & 0.0952 & 0.0476 & 0.0333 & 0.0667 & 0.2333 \\ 0.1569 & 0.1765 & 0.1905 & 0.0952 & 0.0476 & 0.2000 & 0.0333 & 0.1000 \\ 0.1569 & 0.1765 & 0.1905 & 0.0952 & 0.0476 & 0.0667 & 0.2000 & 0.0667 \end{bmatrix}.$$

The steady-state distribution is $\pi = [0.1569 \quad 0.1765 \quad 0.1905 \quad 0.0952 \quad 0.0476 \quad 0.0994 \quad 0.1068 \quad 0.1271]$. This example has a special design so that eight customer types are separated into three groups before generating transition probability matrix: a group of services with negatively correlated traffic streams among group members, and two groups of services with positively correlated traffic streams among group members. The traffic streams between any two types in different groups are independent. This design can be achieved by properly choosing the transition probability so that $\Theta_{i,j} = \pi_j$ and $\Theta_{j,i} = \pi_i$ if we want the traffic stream of type i and j to be independent.

Variance–covariance matrix (Σ) and correlation coefficient matrix (R) can be obtained by Eq. (5.2),

$$\Sigma = \begin{bmatrix} 16.93 & 4.24 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4.24 & 19.58 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 33.30 & -5.56 & -2.03 & 0 & 0 & 0 \\ 0 & 0 & -5.56 & 19.43 & -1.02 & 0 & 0 & 0 \\ 0 & 0 & -2.03 & -1.02 & 9.48 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 11.61 & 0.91 & 0.90 \\ 0 & 0 & 0 & 0 & 0 & 0.91 & 12.31 & 1.20 \\ 0 & 0 & 0 & 0 & 0 & 0.90 & 1.20 & 15.06 \end{bmatrix},$$

$$R = \begin{bmatrix} 1 & 0.23 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.23 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -0.22 & -0.11 & 0 & 0 & 0 \\ 0 & 0 & -0.22 & 1 & -0.07 & 0 & 0 & 0 \\ 0 & 0 & -0.11 & -0.07 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0.08 & 0.07 \\ 0 & 0 & 0 & 0 & 0 & 0.08 & 1 & 0.09 \\ 0 & 0 & 0 & 0 & 0 & 0.07 & 0.09 & 1 \end{bmatrix}.$$

From the correlation coefficient matrix, we observe that the arrival processes of service 1 and 2 are positive correlated with each other and independent of the rest of the services. The arrival processes of service 3, 4, and 5 are negative correlated with each other and independent of all the rest of the services. The arrival processes of service 6, 7, and 8 behave similarly to those of 1 and 2 but with weaker correlations.

With the above data, we can apply the heuristic algorithm. The resulting assignment and routing policies using queueing model and diffusion approximation model are presented in the next two subsections, respectively.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

7.2.1. Queueing model In the initial step of the algorithm, we assume that all assignments are available on every server and solve for the optimal routing policy. Assuming that $d_{i,k} = 1$ for all i and k, the following initial routing policy matrix is obtained by solving $P(d)$,

$$\alpha^*(d) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0.080 & 0.080 & 0.002 & 0.012 & 0.826 \\ 0.080 & 0.080 & 0.002 & 0.012 & 0.826 \\ 0.080 & 0.080 & 0.002 & 0.012 & 0.826 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

As we expected, we observe that the traffic streams with positive correlations are routed into different servers, while the traffic streams with negative correlations are routed into common servers.

In the next step, we remove all assignments with $\alpha_{.,.} = 0$. The total number of assignments left is 20. In terms of the total expected queue length, the system with these 20 assignments can perform as well as the system with 40 assignments, that is, all assignments being enabled to provide every type of service. Since the desired total number of assignments is $T = 12$, we need to proceed with the algorithm further and remove eight more assignments.

Table 1 shows the removal progress using the algorithm. In this table, 1's mean the assignments are removed in the initial step; 2 in the fifth row and third column means the assignment of service type 5 on server 3 is removed when we proceed with the elimination process (the "while loop" in the Algorithm 2) for the first time; 3 in the forth row and third column means the assignment of service type 4 on server 3 is removed when we proceed with the elimination process for the second time, etc. We repeat the elimination process until the number of assignments left is less than or equal to 12. The zeros on the table mean those assignments remain on the servers after the completion of the heuristic algorithm. Based on this table, we determine the service assignment of the system. The assignment policy out of this heuristic algorithm d_h^* should be to enable server k to provide service type i if and only if it is zero in the row i and column k in Table 1.

Table 1. Example I: removal progress of Heuristic algorithm using queueing model

Service Type	Server				
	1	2	3	4	5
1	1	1	1	0	1
2	1	1	1	1	0
3	0	0	6	0	0
4	9	0	3	5	0
5	8	7	2	4	0
6	1	1	1	1	0
7	1	1	1	0	1
8	1	1	0	1	1

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

Along with the above assignment policy, we determine the routing policy out of the heuristic algorithm:

$$\alpha_h^{**} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0.139 & 0.080 & 0 & 0.023 & 0.758 \\ 0 & 0.119 & 0 & 0 & 0.881 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

The expected queue length under this assignment and routing policy is 32.69.

7.2.2. Diffusion approximation model Assuming that $d_{i,k} = 1$ for all i and k again, another initial routing policy is obtained by solving $\tilde{P}(d)$,

$$\tilde{\alpha}^*(d) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0.080 & 0.080 & 0.002 & 0.012 & 0.826 \\ 0.080 & 0.080 & 0.002 & 0.012 & 0.826 \\ 0.080 & 0.080 & 0.002 & 0.012 & 0.826 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

This initial routing policy comes out to be exactly the same as the initial routing policy obtained by queueing model. Similar to what we did for queueing model, we further proceed with the algorithm and use the Table 2 to show the removal progress.

The assignment policy this heuristic algorithm $\tilde{d}_h^*$ should be to enable server k to provide service type i if and only if it is zero in the row i and column k in Table 2. Along with the

TABLE 2. Example I: removal progress of heuristic algorithm using diffusion approximation model

Service Type	Server				
	1	2	3	4	5
1	1	1	1	0	1
2	1	1	1	1	0
3	0	0	6	0	0
4	9	0	3	5	0
5	8	7	2	4	0
6	1	1	1	1	0
7	1	1	1	0	1
8	1	1	0	1	1

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

above assignment policy, we determine the routing policy:

$$\tilde{\alpha}_h^{**} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0.139 & 0.082 & 0 & 0.023 & 0.756 \\ 0 & 0.115 & 0 & 0 & 0.885 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

The total expected queue length under this routing policy is 32.69. In this example, both models give us the same total expected queue length with same assignment policy and slightly different routing policy.

7.2.3. Convex quadratic model We solve the convex quadratic model P^q directly using MIQP solver CPLEX. Given $T = 12$, we obtain the following optimal routing policy:

$$\alpha^{q*} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0.434 & 0.566 & 0 \\ 0 & 0 & 0 & 0.369 & 0.631 \\ 0.218 & 0 & 0 & 0 & 0.782 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.417 & 0 & 0.583 \end{bmatrix}.$$

The total expected queue length under this routing policy is 36.26.

In order to further compare the three models: queueing model P, diffusion approximation model $\tilde{P}$, and convex quadratic model P^q, we perform the following experiment. Using the same set of parameters given above, we solve P, $\tilde{P}$, and P^q to find the optimal static assignment and routing policies. (We use heuristic algorithm for P and $\tilde{P}$, but MIQP for P^q.) Under each of these assignment and routing policies, we compute the expected queue length as explained in Section 3. We summarize the results in Table 3 and Figure 1. The total expected queue lengths with $T = 8$ are not presented because they do not satisfy the stability condition.

We have several key observations from this numerical experiment. The heuristic algorithm provides consistent solutions between queueing model and diffusion approximation model for different number of assignments (T). Owing to the non-convex objective functions, solving nonlinear models using the heuristic algorithm still takes thousands of seconds, even though the diffusion approximation model only takes less than half of the time taken by the queueing model (1,983 versus 5,100 seconds). On the other hand, solving the convex quadratic model only takes a few seconds. For most of the cases, the solutions by convex quadratic model are not as good as the nonlinear model but are in a reasonable range. When T is quite small, the convex quadratic model even provides a better solution than nonlinear models. This is because the heuristic algorithm is not guaranteed to produce the optimal solution.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

TABLE 3. Example I: Total expected queue length under assignment and routing policy derived by solving queueing model P, diffusion approximation model $\tilde{P}$, and convex quadratic model P^q

Number of Assignments (T)	9	10	11	12	13	14	15	16	17	18	19
$\Psi(d_h^*, \alpha_h^{**})$	55.49	33.01	32.69	32.63	32.59	32.57	32.57	32.56	32.56	32.56	32.56
$\Psi(\tilde{d}_h^*, \tilde{\alpha}_h^{**})$	55.49	33.01	32.69	32.63	32.59	32.57	32.57	32.56	32.56	32.56	32.56
$\Psi(d^q, \alpha^q)$	44.44	36.02	35.17	36.26	36.42	36.21	36.28	37.02	38.20	37.16	37.30
Number of Assignments (T)	20	21	22	23	24	25	26	27	28	29	30
$\Psi(d_h^*, \alpha_h^{**})$	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56
$\Psi(\tilde{d}_h^*, \tilde{\alpha}_h^{**})$	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56
$\Psi(d^q, \alpha^q)$	37.44	37.32	37.63	37.60	37.61	37.53	37.66	37.58	37.51	37.42	37.33
Number of Assignments (T)	31	32	33	34	35	36	37	38	39	40	
$\Psi(d_h^*, \alpha_h^{**})$	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	
$\Psi(\tilde{d}_h^*, \tilde{\alpha}_h^{**})$	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	32.56	
$\Psi(d^q, \alpha^q)$	37.32	37.33	37.34	37.34	37.34	37.34	37.34	37.34	37.34	37.30	

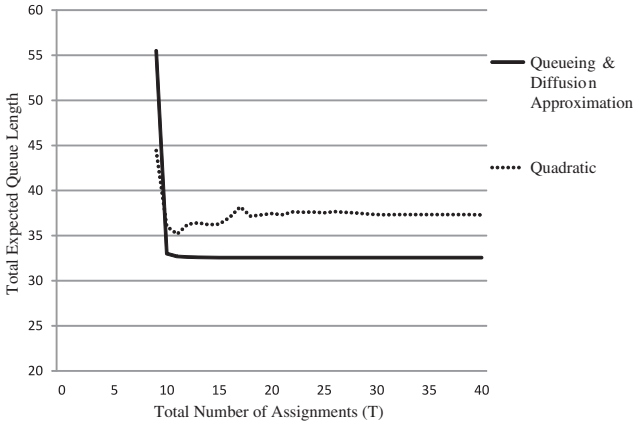

FIGURE 1. Example I: total expected queue length under assignment and routing policy derived by solving queueing model P, diffusion approximation model $\tilde{P}$, and convex quadratic model P^q.

7.3. Numerical Example II: Virtual Computing Laboratory (VCL)

We consider another example with five servers ($M = 5$) and eight types of services ($N = 8$). Here, the system parameters such as the overall arrival rate, the transition probability matrix of the DTMC determining the customer class, and the service time distributions are estimated from a real data set. This data set is an access log file obtained from the VCL in the University of North Carolina at Chapel Hill. The VCL provides access for researchers and students to a virtual computer environment that can be used to access software applications. The data set contains about ten thousand service requests between August and November 2012, each with the type of service requested, the time of arrival, the time of service beginning, and the time of departure.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

We assume that the overall customer arrival process is a PP with rate $\lambda = 73$. This arrival rate is estimated by mean arrival rate between 9 am and 5 pm. This specific time interval is chosen because the data show that the arrival rate is much higher during the daytime. Obviously the overall performance of system is mainly determined by how well the system performs during these peak hours. The transition probability matrix of the DTMC is estimated from the sequence of service requests:

$$\Theta = \begin{bmatrix} 0.3210 & 0.3310 & 0.0160 & 0.1020 & 0.0270 & 0.0280 & 0.0900 & 0.0850 \\ 0.1310 & 0.6550 & 0.0140 & 0.0610 & 0.0130 & 0.0140 & 0.0520 & 0.0600 \\ 0.1330 & 0.3910 & 0.1330 & 0.0570 & 0.0480 & 0.0190 & 0.1050 & 0.1140 \\ 0.2450 & 0.3510 & 0.0120 & 0.1630 & 0.0330 & 0.0310 & 0.0760 & 0.0890 \\ 0.1460 & 0.4070 & 0.0310 & 0.1540 & 0.0540 & 0.0310 & 0.0770 & 0.1000 \\ 0.2180 & 0.3100 & 0.0070 & 0.0770 & 0.0350 & 0.1690 & 0.0990 & 0.0850 \\ 0.2120 & 0.3300 & 0.0160 & 0.0710 & 0.0260 & 0.0280 & 0.2220 & 0.0950 \\ 0.1980 & 0.3780 & 0.0200 & 0.0980 & 0.0220 & 0.0120 & 0.1040 & 0.1680 \end{bmatrix}.$$

The steady-state distribution out of the above transition probability is

$$\pi = [0.1913 \quad 0.5008 \quad 0.0171 \quad 0.0834 \quad 0.0211 \quad 0.0231 \quad 0.0819 \quad 0.0812],$$

and the correlation coefficient matrix (R) is

$$R = \begin{bmatrix} 1 & -0.21 & -0.02 & 0.08 & 0.01 & 0.04 & 0.04 & 0.03 \\ -0.21 & 1 & -0.04 & -0.13 & -0.07 & -0.09 & -0.15 & -0.11 \\ -0.02 & -0.04 & 1 & -0.02 & 0.04 & -0.01 & 0.01 & 0.02 \\ 0.08 & -0.13 & -0.02 & 1 & 0.06 & 0.02 & -0.01 & 0.03 \\ 0.01 & -0.07 & 0.04 & 0.06 & 1 & 0.03 & 0.01 & 0.02 \\ 0.04 & -0.09 & -0.01 & 0.02 & 0.03 & 1 & 0.03 & -0.01 \\ 0.04 & -0.15 & 0.01 & -0.01 & 0.01 & 0.03 & 1 & 0.05 \\ 0.03 & -0.11 & 0.02 & 0.03 & 0.02 & -0.01 & 0.05 & 1 \end{bmatrix}.$$

In this example, we assume that service time is exponentially distributed with rate dependent on service type, which is estimated by mean service rate of each type from the data. The service rate we use is

$$[\mu_{1,k}, \mu_{2,k}, \ldots, \mu_{N,k}] = [12 \quad 20 \quad 20 \quad 50 \quad 12 \quad 12 \quad 20 \quad 20] \quad \forall k.$$

We round the service rates to the choices of three different rates and scale them up proportionally for the ease of application with queueing model. This is not unrealistic because the data show some similarity of service time distributions between different service types.

We apply the heuristic algorithm with queueing model and diffusion approximation model. We do not include the convex quadratic model in the example since it requires the service rates to be independent of service types. For queueing model, assuming $d_{i,k} = 1$ for all i and k, we obtain the following initial routing policy matrix by solving $P(d)$,

$$\alpha^*(d) = \begin{bmatrix} 0.432 & 0 & 0 & 0 & 0.568 \\ 0.119 & 0.304 & 0.303 & 0.274 & 0 \\ 0 & 0.328 & 0.331 & 0.342 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0.299 & 0 & 0 & 0 & 0.701 \\ 0.344 & 0.132 & 0.132 & 0 & 0.392 \\ 0 & 0.340 & 0.341 & 0.320 & 0 \\ 0 & 0.392 & 0.392 & 0.217 & 0 \end{bmatrix}.$$

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

TABLE 4. Example II: total expected queue length under assignment and routing policy derived by solving Queueing model P and diffusion approximation model $\tilde{P}$

Number of Assignments (T)	11	12	13	14	15	16	17	18	19	20
$\Psi(d_h^*, \alpha_h^{**})$	26.78	22.59	22.48	22.43	22.38	22.34	22.33	22.32	22.31	22.31
$\Psi(\tilde{d}_h^*, \tilde{\alpha}_h^{**})$	26.78	22.59	22.54	22.46	22.41	22.35	22.36	22.35	22.34	22.36
Number of Assignments (T)	21	22	23	24	25	26	27	28	29	30
$\Psi(d_h^*, \alpha_h^{**})$	22.30	22.30	22.30	22.30	22.30	22.30	22.30	22.30	22.30	22.30
$\Psi(\tilde{d}_h^*, \tilde{\alpha}_h^{**})$	22.36	22.37	22.36	22.36	22.36	22.36	22.36	22.36	22.36	22.36
Number of Assignments (T)	31	32	33	34	35	36	37	38	39	40
$\Psi(d_h^*, \alpha_h^{**})$	22.30	22.30	22.30	22.30	22.30	22.30	22.30	22.30	22.30	22.30
$\Psi(\tilde{d}_h^*, \tilde{\alpha}_h^{**})$	22.36	22.36	22.36	22.36	22.36	22.36	22.36	22.36	22.36	22.36

Similarly for diffusion approximation model, assuming $d_{i,k} = 1$ for all i and k, we obtain the following initial routing policy matrix by solving $\tilde{P}(d)$,

$$\tilde{\alpha}^*(d) = \begin{bmatrix} 0.500 & 0 & 0 & 0 & 0.500 \\ 0.049 & 0.300 & 0.300 & 0.300 & 0.049 \\ 0 & 0.333 & 0.334 & 0.334 & 0 \\ 0 & 0.333 & 0.334 & 0.333 & 0 \\ 0.500 & 0 & 0 & 0 & 0.500 \\ 0.500 & 0 & 0 & 0 & 0.500 \\ 0 & 0.334 & 0.333 & 0.333 & 0 \\ 0 & 0.333 & 0.333 & 0.334 & 0 \end{bmatrix}.$$

Unlike in Example I, the initial routing policy matrices from two models are not exactly the same, but they still have some similar structures. These two routing policy matrices have lots of common zero entries, which further confirms our conjecture that it is optimal to separate traffic streams to different servers if they are not positively correlated. However, due to the mixed correlation structure in this example, this effect is not as obvious as in Example I.

We then solve P and $\tilde{P}$ using heuristic algorithm for P and $\tilde{P}$ with all the possible T value. Under each of these assignment and routing policies, we compute the expected queue length as explained in Section 3. We summarize the results in Table 4 and Figure 2. The total expected queue lengths with $T = 8, 9, 10$ are not presented because they do not satisfy the stability condition.

Similar to our observations from the previous example, the solutions from two models are not exactly the same, but the resulting expected queue lengths are quite close to each other. In addition, solving diffusion approximation model takes only about one-third of the time that solving queueing model does (5,892 versus 17,812 s).

8. CONCLUSION AND FUTURE STUDY

In this paper, we provide schemes for determining the assignment and routing policies for a service center. First, we formulate the problem as a mixed integer nonlinear programming problem aiming to minimize the system congestion performance measures. Second,

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

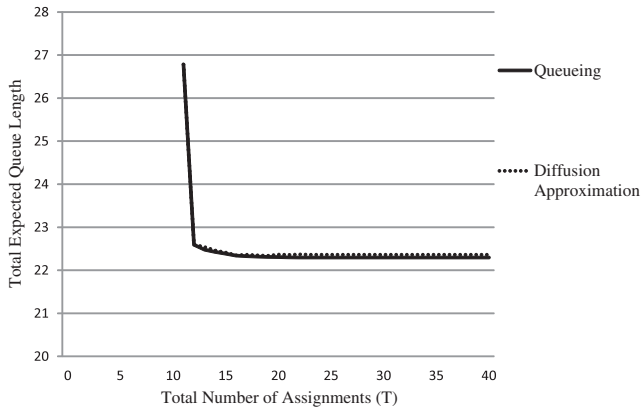

FIGURE 2. Example II: total expected queue length under assignment and routing policy derived by solving queueing model P and diffusion approximation model $\tilde{P}$.

we introduce reasonable performance measures in three models: queueing model, diffusion approximation model, and convex quadratic model. These three models all provide solutions taking account the autocorrelations within traffic steams and correlations between different traffic streams. The queueing model produces expected queue length as performance measure by a matrix analytic method, which is accurate but time consuming. To improve the efficiency, we study the diffusion approximation model which gives a closed form expression of approximated expected queue length, which is easy to compute, but is non-convex. We then define a convex performance measure that is easier to optimize and yields policies that perform near optimally.

As we observed from the numerical examples, the queueing model and diffusion approximation model take much longer time to solve but result in solutions with better performance (smaller total expected queue lengths). When the convex quadratic model is applicable, it provides the most efficient method of deriving assignment and routing policies of acceptable performance. Finally, we develop a greedy heuristic algorithm to increase the scalability. Comparing the numerical results of the queueing model and the diffusion approximation model, we observe that the assignment policies obtained by both models are essentially the same. Hence, using diffusion approximation model can be more efficient without sacrificing system performance.

For the future study, we think it would be interesting to investigate the effect of the correlation structure on the assignment and routing policies. Another interesting direction is to look at how one can reduce the required number of servers while satisfying a given quality of service requirement by taking account the correlations between different traffic streams into design.

Acknowledgements

The authors would like to thank the referee for many valuable suggestions that resulted in considerable improvement of the presentation of this paper.

References

1. Adan, I.J.B.F. & Kulkarni, V.G. (2003). Single-server queue with Markov-dependent inter-arrival and service times. *Queueing Systems* 45: 113–134.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

2. Andradóttir S., Ayhan, H. & Down, D.G. (2001). Server assignment policies for maximizing the steady-state throughput of finite queueing systems. *Management Science* 47(10): 1421–1439.
3. Andradóttir S., Ayhan, H. & Down, D.G. (2003). Dynamic server allocation for queueing networks with flexible servers. *Operations Research* 51(6): 952–968.
4. Anselmi, J., Amaldi, E. & Cremonesi, P. (2008). Service consolidation with end-to-end response time constraints. In *Proceedings of the 2008 34th Euromicro Conference Software Engineering and Advanced Applications*, SEAA '08, pp. 345–352, Washington, DC, USA, IEEE Computer Society.
5. Anselmi, J., Cremonesi, P. & Amaldi, E. (2009). On the consolidation of data-centers with performance constraints. In *Architectures for Adaptive Software Systems*, vol. 5581 of *Lecture Notes in Computer Science* (R. Mirandola, I. Gorton & C. Hofmeister, Eds.), Berlin/Heidelberg: Springer, pp. 163–176.
6. Bassamboo, A., Harrison, J.M. & Zeevi, A. (2006). Design and control of a large call center: asymptotic analysis of an LP-based method. *Operations Research* 54(3): 419–435.
7. Bichler, M., Setzer, T. & Speitkamp, B. (2006). Capacity planning for virtualized servers. Presented at Workshop on Information Technologies and Systems (WITS), Milwaukee, Wisconsin, USA.
8. Borst, S.C. (1995). Optimal probabilistic allocation of customer types to servers. *SIGMETRICS Performance Evaluation Review* 23(1): 116–125
9. Buzacott, J.A. & Shanthikumar, J.G. (1992). Design of manufacturing systems using queueing models. *Queueing Systems* 12(1): 135–213.
10. Chen, Y., Das, A., Qin, W., Sivasubramaniam, A., Wang, Q. & Gautam, N. (2005). Managing server energy and operational costs in hosting centers. *SIGMETRICS Performance Evaluation Review* 33: 303–314.
11. de Véricourt F. & Jennings, O.B. (2011). Nurse staffing in medical units: a queueing perspective. *Operations Research* 59(6): 1320–1331.
12. Good, I.J. (1961). The frequency count of a Markov chain and the transition to continuous time. *The Annals of Mathematical Statistics* 32(1): 41–48.
13. Green, L. (2006). Queueing analysis in healthcare. In *Patient Flow: Reducing Delay in Healthcare Delivery*. volume 91 of *International Series in Operations Research & Management Science*, US: Springer, pp. 281–307.
14. Guo, X., Lu, Y. & Squillante, M.S. (2004). Optimal probabilistic routing in distributed parallel queues. *SIGMETRICS Performance Evaluation Review* 32(2): 53–54.
15. Gurvich, I., Armony, M. & Mandelbaum, A. (2008). Service-level differentiation in call centers with fully flexible servers. *Management Science* 54(2): 279–294.
16. Gurvich, I. & Whitt, W. (2010). Service-level differentiation in many-server service systems via queue-ratio routing. *Operations Research* 58(2): 316–328.
17. Harrison, J.M. & Zeevi, A. (2005). A method for staffing large call centers based on stochastic fluid models. *Manufacturing & Service Operations Management* 7(1): 20–36.
18. Jaumard, B., Semet, F. & Vovor, T. (1998). A generalized linear programming model for nurse scheduling. *European Journal of Operational Research* 107(1): 1–18.
19. Johnson, D.S., Demers A., Ullman, J.D., Garey, M.R. & Graham, R.L. (1974). Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing* 3(4): 299–325.
20. Kulkarni, V.G. (1995). *Modeling and Analysis of Stochastic Systems*. Boca Raton: Chapman & Hall/CRC.
21. Kwak, N.K. & Lee, C. (1997). A linear goal programming model for human resource allocation in a health-care organization. *Journal of Medical Systems* 21: 129–140.
22. Li, T.-H. (2007). A statistical framework of optimal workload consolidation with application to capacity planning for on-demand computing. *Journal of the American Statistical Association* 102(479): 841–855.
23. Sethuraman, J. & Squillante, M.S. (1999). Optimal stochastic scheduling in multiclass parallel queues. *SIGMETRICS Performance Evaluation Review* 27(1): 93–102.
24. Shanthikumar, J.G. & Xu, S.H. (1997). Asymptotically optimal routing and service rate allocation in a multiserver queueing system. *Operations Research* 45(3): 464–469.
25. Speitkamp, B. & Bichler, M. (2010). A mathematical programming approach for server consolidation problems in virtualized data centers. *IEEE Transactions on Services Computing* 3: 266–278.
26. Tekin, S., Andradóttir, S. & Down, D. (2012). Dynamic server allocation for unstable queueing networks with flexible servers. *Queueing Systems* 70: 45–79.
27. Vogels, W. (2008). Beyond server consolidation. *Queue* 6(1): 20–26.
28. Wallace, R.B. & Whitt, W. (2005). A staffing algorithm for call centers with skill-based routing. *Manufacturing & Service Operations Management* 7(4): 276–294.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

29. Wang, Y.-T. & Morris, R.J.T. (1985). Load sharing in distributed systems. *IEEE Transactions on Computers* C-34(3): 204–217.
30. Whitt, W. (1982). Refining diffusion approximations for queues. *Operations Research Letters* 1(5): 165–169.
31. Whitt, W. (2006). A multi-class fluid model for a contact center with skill-based routing. *AEU — International Journal of Electronics and Communications* 60(2): 95–102.
32. Yankovic, N. & Green, L.V. (2011). Identifying good nursing levels: a queuing approach. *Operations Research* 59(4): 942–955.

APPENDIX A. PROOFS OF THEOREMS

A.1. Proof of Theorem 3

PROOF: Recall that $\Sigma_{i,j} = \lim_{t\to\infty} \frac{\mathrm{Cov}(A_i(t),A_j(t))}{t}$. By (8) from Good [12], we know that

$$E\left[A_i(t)A_j(t)\,|A(t)=r\right] = \mathbb{1}_{\{i=j\}}\pi_i r + 2\binom{r}{2}\pi_i\pi_j + r\pi_i[(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}]_{i,j} + r\pi_j[(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}]_{j,i} + O(1), \tag{A.1}$$

and hence

$$\begin{aligned} E[A_i(t)A_j(t)] &= E\left[E\left(A_i(t)A_j(t)\,|A(t)\right)\right] \\ &= E\left(A(t)\right)\{\mathbb{1}_{\{i=j\}}\pi_i + \pi_i[(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}]_{i,j} \\ &\quad + \pi_j[(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}]_{j,i}\} + \pi_i\pi_j E\left[A(t)\left(A(t)-1\right)\right] + O(1) \\ &= \lambda t\{\mathbb{1}_{\{i=j\}}\pi_i + \pi_i[(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}]_{i,j} \\ &\quad + \pi_j[(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}]_{j,i}\} + \pi_i\pi_j(\lambda t)^2 + O(1). \end{aligned} \tag{A.2}$$

Then we can derive $\Sigma_{i,j}$,

$$\begin{aligned} \Sigma_{i,j} &= \lim_{t\to\infty}\frac{\mathrm{Cov}[A_i(t),A_j(t)]}{t} = \lim_{t\to\infty}\frac{E[A_i(t)A_j(t)] - E[A_i(t)]E[A_j(t)]}{t} \\ &= \lambda\{\mathbb{1}_{\{i=j\}}\pi_i + \pi_i[(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}]_{i,j} \\ &\quad + \pi_j[(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}]_{j,i}\}, \end{aligned} \tag{A.3}$$

and hence

$$\begin{aligned} \Sigma = \lambda\{&\,\mathrm{diag}[\pi] + \mathrm{diag}[\pi][(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}] \\ &+ [(\Theta - e_N\pi)(I_N - \Theta + e_N\pi)^{-1}]'\,\mathrm{diag}[\pi]\}. \end{aligned} \tag{A.4}$$

■

A.2. Proof of Theorem 4

PROOF: Given that

$$E(\tilde{U}_{n,k}) = \frac{1}{\lambda\pi\alpha_k}, \tag{A.5}$$

we know that

$$\lim_{t\to\infty}\frac{E(\tilde{B}_k(t))}{t} = \lambda\pi\alpha_k \tag{A.6}$$

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

by the elementary renewal theorem. This shows that Eq. (5.5) holds since we also know that

$$\lim_{t\to\infty}\frac{E(B_k(t))}{t}=\lambda\pi\alpha_k. \tag{A.7}$$

from Eq. (3.2).

By Theorem 8.7 of Kulkarni [20], we know that

$$\lim_{t\to\infty}\frac{\mathrm{Var}(\tilde{B}_k(t))}{t}=\frac{\mathrm{Var}(\tilde{U}_{n,k})}{(E(\tilde{U}_{n,k}))^3}=\lambda\pi\alpha_k+\alpha_k'(\Sigma-\lambda\,\mathrm{diag}[\pi])\alpha_k. \tag{A.8}$$

To show that Eq. (5.6) holds, we check

$$\begin{aligned}
\lim_{t\to\infty}\frac{\mathrm{Var}(B_k(t))}{t}&=\lim_{t\to\infty}\frac{\mathrm{Var}\left(\sum_{i=1}^N Y_{i,k}(t)\right)}{t}=\lim_{t\to\infty}\frac{\mathrm{Var}[\sum_{i=1}^N \mathrm{Bin}(\alpha_{i,k},A_i(t))]}{t}\\
&=\lim_{t\to\infty}\frac{\sum_{i=1}^N \mathrm{Var}[\mathrm{Bin}(\alpha_{i,k},A_i(t))]}{t}\\
&\quad+\lim_{t\to\infty}\frac{\sum_{i=1}^N\sum_{j=1,i\neq j}^N \mathrm{Cov}[\mathrm{Bin}(\alpha_{i,k},A_i(t)),\mathrm{Bin}(\alpha_{j,k},A_j(t))]}{t}\\
&=\lim_{t\to\infty}\frac{\sum_{i=1}^N \mathrm{Var}\{E[\mathrm{Bin}(\alpha_{i,k},A_i(t))|A_i(t)]\}+\sum_{i=1}^N E\{\mathrm{Var}[\mathrm{Bin}(\alpha_{i,k},A_i(t))|A_i(t)]\}}{t}\\
&\quad+\lim_{t\to\infty}\frac{\sum_{i=1}^N\sum_{j=1,i\neq j}^N \mathrm{Cov}\{E[\mathrm{Bin}(\alpha_{i,k},A_i(t))|A_i(t),A_j(t)],E[\mathrm{Bin}(\alpha_{j,k},A_j(t))|A_i(t),A_j(t)]\}}{t}\\
&\quad+\lim_{t\to\infty}\frac{\sum_{i=1}^N\sum_{j=1,i\neq j}^N E\{\mathrm{Cov}[\mathrm{Bin}(\alpha_{i,k},A_i(t)),\mathrm{Bin}(\alpha_{j,k},A_j(t))|A_i(t),A_j(t)]\}}{t}\\
&=\sum_{i=1}^N\left[\alpha_{i,k}^2\lim_{t\to\infty}\frac{\mathrm{Var}(A_i(t))}{t}+\alpha_{i,k}(1-\alpha_{i,k})\lim_{t\to\infty}\frac{E(A_i(t))}{t}\right]\\
&\quad+\sum_{i=1}^N\sum_{j=1,i\neq j}^N\alpha_{i,k}\alpha_{j,k}\lim_{t\to\infty}\frac{\mathrm{Cov}(A_i(t),A_j(t))}{t}\\
&=\lambda\pi\alpha_k+\alpha_k'(\Sigma-\lambda\,\mathrm{diag}[\pi])\alpha_k.
\end{aligned} \tag{A.9}$$

This completes the proof of Theorem 4. ■

A.3. Proof of Theorem 5

Proof: Let X_n be the type of nth customer arriving to queue k. The expectation of service times of queue k is given by

$$\lim_{n\to\infty}E(V_{n,k})=\lim_{n\to\infty}E[E(V_{n,k}|X_n)]=\sum_{i=1}^N\frac{\tau_{i,k}\pi_i\alpha_{i,k}}{\sum_{i=1}^N\pi_i\alpha_{i,k}}=\frac{\tau_k\,\mathrm{diag}[\pi]\alpha_k}{\pi\alpha_k}, \tag{A.10}$$

and the variance of service times of queue k is given by

$$\begin{aligned}
\lim_{n\to\infty}\mathrm{Var}(V_{n,k})&=\lim_{n\to\infty}\{E[\mathrm{Var}(V_{n,k}|X_n)]+\mathrm{Var}[E(V_{n,k}|X_n)]\}\\
&=\sum_{i=1}^N\frac{\sigma_{i,k}^2\pi_i\alpha_{i,k}}{\sum_{i=1}^N\pi_i\alpha_{i,k}}+\sum_{i=1}^N\frac{\tau_{i,k}^2\pi_i\alpha_{i,k}}{\sum_{i=1}^N\pi_i\alpha_{i,k}}-\left(\frac{\tau_k\,\mathrm{diag}[\pi]\alpha_k}{\pi\alpha_k}\right)^2,\\
&=\frac{(\sigma_k^2+\tau_k^2)\,\mathrm{diag}[\pi]\alpha_k}{\pi\alpha_k}-\left(\frac{\tau_k\,\mathrm{diag}[\pi]\alpha_k}{\pi\alpha_k}\right)^2.
\end{aligned} \tag{A.11}$$

This shows that both Eqs. (5.9) and (5.10) hold. ■

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

A.4. Proof of Theorem 6

PROOF: To prove Theorem 6, we need to show that the sum of expected queue lengths over all servers is minimized when the traffic intensity is the same for every server under the assumptions in the theorem. Let λ_k be the arrival rate to server k and then total arrival rate $\lambda = \sum_{k=1}^{M} \lambda_k$. Since we assume that the service time distributions are all the same, we can let μ be the service rate and C_s^2 be the squared coefficient of variation of service times on every server. We can further define traffic intensity of server k to be

$$\rho_k = \lambda_k/\mu = \lambda\pi\alpha_k/\mu,$$

and traffic intensity of whole system to be

$$\rho = \lambda/(M\mu).$$

We will show the result for queueing model. The same argument can apply to diffusion approximation model. Since each queue k is an $M/G/1$ queue, one may write down $L_k(\alpha_k)$, the expected queue length of queue k, by Pollaczek–Khinchin formula. The total expected queue length minimization problem can be written as

$$\min \quad \sum_{k=1}^{M} L_k(\alpha_k) = \sum_{k=1}^{M} \left[\rho_k + \frac{\rho_k^2(1+c_{s_k}^2)}{2(1-\rho_k)}\right], \tag{A.12}$$

$$\text{s.t.} \quad \lambda = \sum_{k=1}^{M} \lambda_k \quad \text{or} \quad M\rho = \sum_{k=1}^{M} \rho_k, \tag{A.13}$$

$$0 \le \rho_k \le 1 \quad \forall k \in \{1, \ldots, M\}. \tag{A.14}$$

Taking derivatives of $L_k(\alpha_k)$ with respect to ρ_k, we have

$$\frac{dL_k(\alpha_k)}{d\rho_k} = 1 + \frac{\rho_k(1+c_{s_k}^2)(2-\rho_k)}{2(1-\rho_k)^2}, \tag{A.15}$$

$$\frac{d^2L_k(\alpha_k)}{d\rho_k^2} = \frac{1+c_{s_k}^2}{(1-\rho_k)^3} > 0 \quad (\text{given } 0 \le \rho_k < 1). \tag{A.16}$$

Hence, we know that $L_k(\alpha_k)$ is a convex function of ρ_k and so is $\sum_{k=1}^{M} L_k(\alpha_k)$. We can integrate the constraint $M\rho = \sum_{k=1}^{M} \rho_k$ into objective function and rewrite the problem as

$$\begin{aligned}\min \quad \sum_{k=1}^{M} L_k(\alpha_k) &= \sum_{k=1}^{M-1} \left[\rho_k + \frac{\rho_k^2(1+c_{s_k}^2)}{2(1-\rho_k)}\right] + \left(M\rho - \sum_{k=1}^{M-1} \rho_k\right) + \frac{(M\rho - \sum_{k=1}^{M-1}\rho_k)^2(1+c_{s_k}^2)}{2(1-M\rho+\sum_{k=1}^{M-1}\rho_k)} \\ &= M\rho + \sum_{k=1}^{M-1} \frac{\rho_k^2(1+c_{s_k}^2)}{2(1-\rho_k)} + \frac{(M\rho - \sum_{k=1}^{M-1}\rho_k)^2(1+c_{s_k}^2)}{2(1-M\rho+\sum_{k=1}^{M-1}\rho_k)},\end{aligned} \tag{A.17}$$

subject to $0 \le \rho_k \le 1$, $\forall k$. Taking the first-order partial derivative of objective function with respect to ρ_k, $\forall 1 \le k \le M-1$, we have

$$\frac{\partial \sum_{k=1}^{M} L_k(\alpha_k)}{\partial \rho_k} = \left(\frac{1+c_{s_k}^2}{2}\right)\left[\frac{2\rho_k - \rho_k^2}{(1-\rho_k)^2} - \frac{2(M\rho - \sum_{k=1}^{M-1}\rho_k) - (M\rho - \sum_{k=1}^{M-1}\rho_k)^2}{(1-M\rho+\sum_{k=1}^{M-1}\rho_k)^2}\right]. \tag{A.18}$$

Setting $\frac{\partial \sum_{k=1}^{M} L_k(\alpha_k)}{\partial \rho_k} = 0$ for all $1 \le k \le M-1$ and knowing that $0 \le \rho_k < 1$, we have

$$\frac{2\rho_k - \rho_k^2}{(1-\rho_k)^2} = \frac{2(M\rho - \sum_{k=1}^{M-1}\rho_k) - (M\rho - \sum_{k=1}^{M-1}\rho_k)^2}{(1-M\rho+\sum_{k=1}^{M-1}\rho_k)^2}, \quad \forall k \in \{1, \ldots, M-1\}, \tag{A.19}$$

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

and $\rho_1 = \cdots = \rho_{M-1} = \rho$ is a solution that satisfies the equation above. Hence, we can conclude that $\rho_1 = \cdots = \rho_M = \rho$ is an optimal solution for queueing model because it satisfies all constraints, have all first-order partial derivatives equal to zero, and the objective function is convex.

We may follow the same fashion to prove the result of diffusion approximation model. The details are omitted. ■

A.5. Proof of Theorem 8

PROOF: Assuming that $d_{i,k} = 1$ for all i and k, P^q can be reduced to
Problem $P^{q'}$

$$\min \quad \sum_{k=1}^{M} \hat{Q}_k(\alpha_k) = \sum_{k=1}^{M} [\alpha_k' \hat{\Sigma} \alpha_k + (\lambda \pi \alpha_k - \mu_k)^2], \tag{A.20}$$

$$\text{s.t.} \quad \sum_{k=1}^{M} \alpha_{i,k} = 1, \qquad \forall i \in \{1, \ldots, N\}, \tag{A.21}$$

$$\alpha_{i,k} \geq 0, \qquad \forall i \in \{1, \ldots, N\},\ k \in \{1, \ldots, M\}. \tag{A.22}$$

This is a quadratic programming problem. Let e_N be an N-vector whose elements are all ones. We rewrite the $P^{q'}$ in a standard form of quadratic programming problem,
Problem $P^{q''}$

$$\min \quad \frac{1}{2} x' H x + c' x, \text{ where } H = \begin{bmatrix} \hat{\Sigma} + \lambda^2 \pi' \pi & & \\ & \ddots & \\ & & \hat{\Sigma} + \lambda^2 \pi' \pi \end{bmatrix}_{MN \times MN}$$

$$\text{and } c = \begin{bmatrix} -\lambda \mu_1 \pi' \\ \vdots \\ -\lambda \mu_M \pi' \end{bmatrix}_{MN}, \tag{A.23}$$

$$\text{s.t.} \quad Ax \geq b, \text{ where } A = I_{MN} \text{ and } b = [0, \ldots, 0]'_{MN}, \tag{A.24}$$

$$Bx = d, \text{ where } B = [I_N, \ldots, I_N]_{N \times MN} \text{ and } d = e_N, \tag{A.25}$$

$$x = [\alpha_1, \ldots, \alpha_M]_{MN} \geq 0, \text{ where } \alpha_k = [\alpha_{1,k}, \ldots, \alpha_{N,k}]' \text{ for all } k. \tag{A.26}$$

We first need to show that $\hat{\Sigma} e_N = \lambda \pi'$. By definition,

$$[\Sigma e_N]_i = \sum_{j=1}^{N} \text{Cov}(A_i(1), A_j(1)) = \text{Cov}(A_i(1), A(1)). \tag{A.27}$$

For any i,

$$\begin{aligned} E[A_i(1)A(1)] &= E[E(A_i(1)A(1)|A(1))] = E[E(A_i(1)|A(1))A(1)] = E[\pi_i A(1)^2] \\ &= \pi_i \text{Var}(A(1)) + \pi_i [E(A(1))]^2, \end{aligned} \tag{A.28}$$

and

$$E(A_i(1))E(A(1)) = \pi_i [E(A(1))]^2. \tag{A.29}$$

Hence, we know that

$$\text{Cov}(A_i(1), A(1)) = E[A_i(1)A(1)] - E(A_i(1))E(A(1)) = \pi_i \text{Var}(A(1)) = \lambda \pi_i, \tag{A.30}$$

$$\Rightarrow \quad \hat{\Sigma} e_N = \lambda \pi'. \tag{A.31}$$

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

Let x_{0_j} be the jth element of the column vector $x_0 \in R^{MN}$. Next, we show that Eq. (6.18) is an optimal solution to P^q in the special case, that is,

$$x_{0_j} = \begin{cases} \dfrac{1+\lambda+|L|\mu_k - \sum_{l\in L}\mu_l}{(1+\lambda)|L|}, & \forall k \in L,\ j \in \{(k-1)N+1,\ldots,kN\}, \\ 0, & \text{otherwise}, \end{cases} \tag{A.32}$$

is an optimal solution to $P^{q''}$. We will check that x_0 is an optimal solution by showing that there exist $u_0 \in R^{MN}$ and $v_0 \in R^N$ such that $Hx_0 + c = A'u_0 + B'v_0$, $Ax_0 \geq b$, $u_0 \geq 0$, $Bx_0 = d$, and $< Ax_0 - b, u_0 >= 0$ (Karush–Kuhn–Tucker conditions). Let u_{0_j} be the jth element of the column vector $u_0 \in R^{MN}$. We pick

$$u_{0_j} = \begin{cases} \dfrac{(-\lambda-\lambda^2-|L|\mu_k\lambda+\sum_{l\in L}\mu_l\lambda)\pi_i}{|L|}, & \forall k \notin L,\ i \in \{1,\ldots,N\},\ j = (k-1)N+i, \\ 0, & \text{otherwise}. \end{cases} \tag{A.33}$$

and

$$v_0 = \frac{(\lambda+\lambda^2-\sum_{l\in L}\mu_l\lambda)\pi'}{|L|}. \tag{A.34}$$

Let us verify all the conditions.

1. $Ax_0 = x_0 \geq b$ and $u_0 \geq 0$ because x_0 and u_0 are both non-negative by the definition of L.
2. $$\begin{aligned} Bx_0 &= \left[\sum_{k\notin L} 0 + \sum_{k\in L}\frac{1+\lambda+|L|\mu_k-\sum_{l\in L}\mu_l}{(1+\lambda)|L|}\right]_N \\ &= \left[\frac{(1+\lambda)|L|+|L|\sum_{k\in L}\mu_k-|L|\sum_{l\in L}\mu_l}{(1+\lambda)|L|}\right]_N = e_N = d. \end{aligned} \tag{A.35}$$
3. $\langle Ax_0 - b, u_0\rangle = \langle x_0, u_0\rangle = \sum_{j=1}^{MN} x_{0_j}u_{0_j} = 0$ because x_{0_j} and u_{0_j} are not non-zero at the same time for any j.
4. Let $(Hx_0)_k$ be the column vector with $(k-1)N+1$th to kNth elements of the vector Hx_0. For any $k \in L$,
$$\begin{aligned} (Hx_0)_k &= (\Sigma+\lambda^2\pi'\pi)e_N\left(\frac{1+\lambda+|L|\mu_k-\sum_{l\in L}\mu_l}{(1+\lambda)|L|}\right) \\ &= (\lambda\pi'+\lambda^2\pi')\left(\frac{1+\lambda+|L|\mu_k-\sum_{l\in L}\mu_l}{(1+\lambda)|L|}\right) \\ &= \left(\frac{\lambda+\lambda^2+|L|\mu_k\lambda-\sum_{l\in L}\mu_l\lambda}{|L|}\right)\pi', \end{aligned} \tag{A.36}$$
$$(Hx_0+c)_k = \left(\frac{\lambda+\lambda^2-\sum_{l\in L}\mu_l\lambda}{|L|}\right)\pi' = A'u_0 + B'v_0. \tag{A.37}$$

   For any $k \notin L$,
$$(Hx_0)_k = 0, \tag{A.38}$$
$$(Hx_0+c)_k = \lambda\mu_k\pi' = A'u_0 + B'v_0. \tag{A.39}$$

From the above verification, we can conclude that x_0 is an optimal solution since the objective function is convex quadratic and there exist $u_0 \in R^{MN}$ and $v_0 \in R^N$ that satisfy the Karush–Kuhn–Tucker conditions. ■

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press

APPENDIX B. BACKWARD SELECTION HEURISTIC ALGORITHM

Assuming to solve P, we write down the Backward Selection Heuristic Algorithm as below:

Algorithm 2

Backward Selection Heuristic Algorithm

$t \leftarrow MN$;
for $i = 1 \rightarrow N,\ k = 1 \rightarrow M$ **do**
  $d_{i,k} \leftarrow 1$;
end for
$\alpha \leftarrow \alpha^*(d)$ as defined in *Problem* $P(d)$;
for $i = 1 \rightarrow N,\ k = 1 \rightarrow M$ **do**
  if $\alpha_{i,k} = 0$ **then**
    $d_{i,k} \leftarrow 0$;
  end if
end for
$t \leftarrow \sum_{i=1}^{N} \sum_{k=1}^{M} d_{i,k}$
while $t > T$ **do**
  for $i = 1 \rightarrow N,\ k = 1 \rightarrow M$ **do**
    if $d_{i,k} = 0$ **then**
      $\Psi_{i,k} \leftarrow \infty$;
    else
      $d_{i,k} \leftarrow 0$;
      $\Psi_{i,k} \leftarrow \Psi(d)$ as defined in *Problem* $P(d)$;
      $d_{i,k} \leftarrow 1$;
    end if
  end for
  $(i,k) \leftarrow \arg\min_{i,k} \Psi_{i,k}$;
  $d_{i,k} \leftarrow 0$;
  $t \leftarrow t - 1$;
end while

To solve the *Problem* $\tilde{P}$, one can simply replace $\alpha^*(d)$ and $\Psi(d)$ with $\tilde{\alpha}^*(d)$ and $\tilde{\Psi}(d)$ in the Algorithm 2.

https://doi.org/10.1017/S0269964814000011 Published online by Cambridge University Press