Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds.

PRIORITY-BASED ROUTING WITH STRICT DEADLINES AND SERVER FLEXIBILITY UNDER UNCERTAINTY

Hoda Parvin

Industrial and Operations Engineering
University of Michigan
1205 Beal Ave
Ann Arbor, MI, 48109, USA

Abhijit Bose

IBM T.J. Watson Research Center
19 Skyline Dr
Hawthorne, NY, 10532, USA

Mark P. Van Oyen

Industrial and Operations Engineering
University of Michigan
1205 Beal Ave
Ann Arbor, MI, 48109, USA

ABSTRACT

In this research we present a simulation-based approach to study alternative dynamic assignment policies in an information technology (IT) service delivery environment. Our overarching goal is to find the most *cost-effective* assignment of service requests to cross-trained agents in a large-scale network. We present a novel heuristic algorithm that assigns an analytically described allocation index to each service request that has arrived. It incorporates factors such as variability in agents' capabilities, uncertainty in request inter-arrival times and complex service level agreements (SLA). We investigate the effectiveness of our proposed assignment algorithm using real world data from an IT service environment on a small problem instance. We discuss how the results of this simulation can help improve the terms of service level contracts as well as agent training programs.

1 INTRODUCTION

Recently, the ability to deliver IT services from multiple locations has given rise to an entirely new IT service delivery model. In this model, organizations outsource components of their IT infrastructure operations to one or more service providers, who in turn use a combination of onsite and offsite resources to manage the components on behalf of their clients. To provide these capabilities, service providers have pioneered a new onsite-offsite delivery model called the *Global Delivery Model,* in which a service provider uses a number of delivery centers around the globe to provide services to its clients. The agents at these delivery centers perform a variety of tasks including remote monitoring and management of hardware and software on a 24x7 basis, develop new applications, test configurations, apply security patches, etc. While call centers are often the primary interface between clients and service providers, the delivery centers perform a variety of tasks at different levels of service complexity behind the scenes. Using this model, the clients of the global delivery model can scale their core business operations to match demand without worrying about resources and skills required to manage their IT infrastructure. At the same time, by leveraging local skills, cost structure and process standardization, service providers can ensure high quality of the services performed from different locations.

The critical factor for a service provider to achieve a high service quality in the global delivery model is efficient utilization of the agents and resources available at its delivery centers. The process by which a service provider assigns a service request to an agent at a service delivery center is known as "dispatching". The nature of the IT service centers presents a number of challenges to efficient dispatching, primarily due to variability in agent skills and complexity of service requests. It is difficult to incorporate agent skill variability and service request complexity in the dispatching procedure for several reasons. First, there have been very few studies on quantifying agent skill variability in performing IT infrastructure services. The experience level and skill sets of an agent strongly affect diagnosis and resolution of IT service requests, and therefore it is crucial that service requests are dispatched to the most suitable agents. Second, IT service requests exhibit a wide range of complexity and therefore require varying levels of effort and coordination by the agents.

Although the well-studied call center staffing problem has some similarity with the service-dispatching problem, there are many important differences. In a service request fulfillment system there is no abandonment whereas in a call center environment, requests may leave the system even before their service begins. Furthermore, IT service centers typically serve requests with different levels of priority (or severity) whereas call centers have a more homogenous demand structure.

The main contributions of our study are: (1) a model to address variability in service time and skill level for each agent for each specified request type, (2) a novel dispatching policy that considers the complexity of each service request in addi-

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

tion to the variability in agent service times and skill levels, and (3) investigation of the performance of this proposed approach using data from an IT service center.

2 LITERATURE REVIEW

Determining the number of agents and their skill requirements is a well studied problem in the literature. As demonstrated by Bartholdi III (1981) even in the deterministic case where the demand and supply are fixed, finding the number of agents required for each time period can be a complex task. In real world scenarios, one faces even more complexities due to the stochastic nature of the demand. Another dimension of complexity is taking into account an agent's skill level, where scheduling of different skills is necessary to serve an incoming request. Van Oyen et al. (2001) addressed, how cross training can be aligned with organizational strategies. Their model includes improving performance measures of the system but does not consider fixed deadlines for jobs in the system. Brusco et al. (1998) used integer programming to show that cross training can be a very useful approach when agent skills can be combined in a sequential setting.

Perhaps the most relevant research framework to our problem can be found in the call center literature. Similar to call centers, IT service centers require agents with a variety of skills to handle the arriving requests. However, it is almost impossible to expect that every agent is fully cross-trained for every task. Therefore it is important to route problems to the *best* agents considering their skills and skill levels.

Investigation of different algorithms and methodologies for routing traffic has been done with the purpose of system improvement. Many researchers considered this concept to improve the Quality of Service (QoS). Ma and Steenkiste (1998) proposed a model in order to decrease delay time of calls for a call center as a measure of QoS by finding a feasible path.

Feldman et al. (2008) and Whitt (2006) studied the problem of call center staffing in a complex structure. They investigated the effect of time-varying demand on the performance measures of the system. They also studied variability of the service time.

Finding the most appropriate agent training policy is another important factor which can affect both the stability and the performance of the system. It is important to note that training all agents for all skills is typically too expensive, or rather impossible. A suitable algorithm can approximate the number of agents with the required set of skills. Wallace and Whitt (2005) introduced an algorithm to route problems based on the skill level of the agent. Other algorithms were developed by Sisselman and Whitt (2007). They introduced the concepts of “value-based routing” and “preference-based routing” to the existing skill-based routing algorithm. All of these algorithms are based on simulation and heuristic approaches but none addressed having different priority of requests and strict (or hard) deadlines, both of which are incorporated in our proposed approach.

3 SYSTEM DESCRIPTION

The dispatching system consists of several agents who are assigned to different service request resolution groups. Each resolution group is capable of handling several different types of service requests. Each service request can be characterized by a request type and a priority level. We assume that each service request with a specified type and priority level has a lump sum penalty cost associated with violating its deadline according to a service level agreement (SLA) contracted with the customer. The inter-arrival time distributions are independent (but not necessarily identical) for each request type and priority level. The time required to resolve a request type can vary by agent.

The goal of an efficient dispatching policy is to minimize the total penalty cost of violating deadlines. The differences between a traditional call center problem and the IT service center problem motivate us to study this problem. Here we mention some of these differences. First, in a typical call center, the SLA is solely a function of the waiting time for the customer before the service starts (i.e. waiting time in the queue), whereas in an IT environment, the SLA is concerned with the total time a request spends in the system until the request is resolved. This, in turn, adds another source of uncertainty to the dispatching problem. Second, in a call center, a customer may leave the system before the service starts. However, in a typical IT application, a customer would never leave the system before the request is resolved. Third, the nature of operations in IT environments require a more sophisticated set of skills for agents, whereas agents in a call center generally have a more limited set of skills. In fact, requests that cannot be resolved at the call center are often passed on to the IT service centers. Fourth, IT service requests may be handled by multiple agents simultaneously depending on their complexity, whereas in a call center, each agent typically handles a single call in its entirety or possibly hands of the call in a sequential fashion.

In this research, we study agent-level variability and complexity of the service request to build a dispatching model for IT infrastructure service requests in a single-stage service delivery center. Figure 1 presents an example of dispatching systems commonly found in many service delivery centers. We discuss factors critical to developing an efficient dispatching policy. These factors include an agent's skill level, service time variability, and the penalty cost associated with violating the deadline.

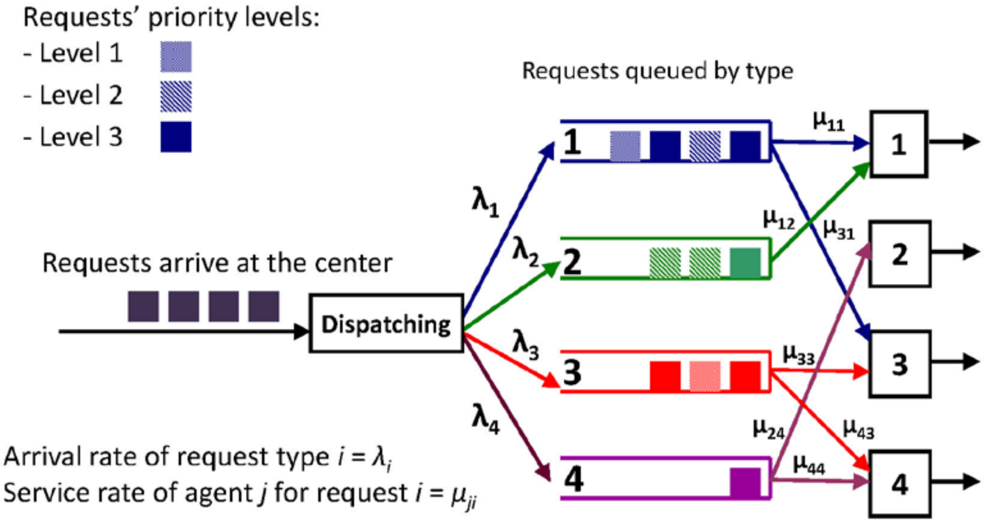

Figure 1: The service request dispatching and resolution process

4 INPUT MODELING

In a typical IT service delivery environment, there is considerable variability among the agents in resolving requests of similar type. This variability stems from experience level (e.g. number of years), certifications, subject-matter expertise, specialized training, and other factors. As a result, some agents are able to diagnose the root causes of a problem faster and more accurately than others, resulting in a shorter mean service restoration time and a smaller standard deviation. There is also a temporal variation in performance of an agent in performing the same task when monitored over time. This can be attributed as an inconsistent performance level and is generally complex to model. In recent years, manufacturing processes have been automated to the point that only random sources of variation are allowed resulting in mostly normal distributions of the process outcomes. However, many IT service management processes such as maintaining servers, patch management, and installation have a high degree of manual involvement. This results in a high agent-induced variability in the process.

Variability has many sources. However, our input analysis shows that the two most important factors that contribute to variability in service resolution time are the complexity level of service requests and their priority level. The complexity level of service requests is illustrated in Figure 2.

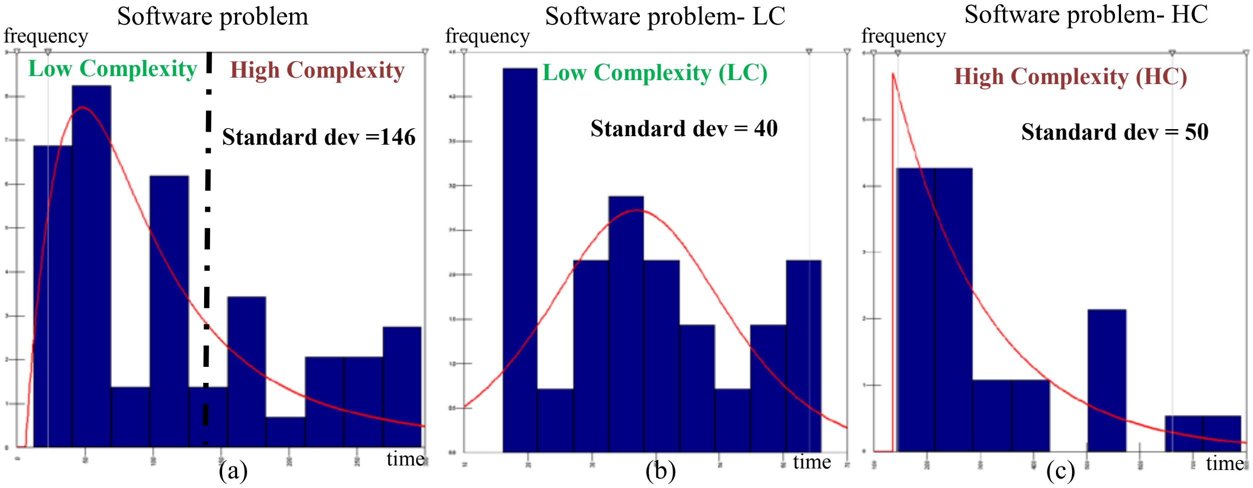

Figure 2: Incorporating task complexity in defining a request type can decrease variability

Figure 2(a) shows a histogram of service time associated with a sampled agent, serving requests categorized as "software problems". Based on this high-level categorization, the service time has a standard deviation of 146. We then refine the categorization further by introducing a complexity level. This way, the “software problem” is broken into two categories: “low-complexity software problems (LC)” and “high-complexity software problems (HC)”. Examples of such problems are restarting an application (LC) versus installing a new application middleware on a server (HC). As shown in Figures 2(b) and 2(c), adding a dimension of complexity dramatically reduces the standard deviation of service time. As a result of this analysis, we redefine the request categorization across all request types in the system to account for the complexity level of service requests. This way, for each request type, at each complexity level, and for each agent we fit a probability distribution to the service time. We note that mapping an incoming service request to its complexity level can be done either manually (by a human dispatcher) or automatically by a classification agent that analyzes the text description of the request and other structured attributes before assigning a complexity class to the request. We describe the key aspects of our simulation model next.

5 SIMULATION MODEL

5.1 Priority-Based Allocation Index

In order to heuristically address the dispatching model described above, we develop a policy to assign each service request to the appropriate agent, based on an *allocation index*. The priority-based allocation index associated with service request *k* with type *i* and priority *s*, if allocating agent *j*, where the remaining time to deadline is *td*, is defined in (1):

$$I_{jis}^{k}(\theta, td) = \frac{m_{ji}(\theta,td)\pi_{is}}{t_{ji}^{\alpha}} \qquad (1)$$

where:

π_{is}= penalty cost of violating the deadline associated with service request *i* with priority s,
t_{ji}^{α} = time required by agent *j* to serve service request type *i* with a confidence level of $1-\alpha$,
td = remaining time to deadline,
$m_{ji}(\theta, td)$= a step function associated with agent *j* and request type *i* to be defined below,
θ = a positive coefficient used to tune the algorithm to an application.

Whenever an agent finishes serving a request (i.e. the system is non-preemptive), we update all the indices for all the requests and the request with the highest index will be assigned to the idle agent. In order to develop this index, we calculate the average service rate at which agent *j* can solve the type associated with the service request *i* (μ_{ji}) from historical data. However, it is important to note that the average service times do not capture the inherent variability of the service time. In the case of IT applications, this is even more important due to non-normal distributions that are a consequence of manually-performed steps, as explained by Pyzde (1995). Therefore, we need to include another measure that better represents the service time variability.

An appropriate measure of variability can be defined based on the confidence level at which an agent can serve a request before its deadline. Let t_{ji}^{α} denote the time it takes for agent *j* to serve request type *i* with probability 1- α (see (2)). Based on this equation, the values of t_{ji}^{α} are calculated a priori.

$$P_{ji}\left(T \le t_{ji}^{\alpha}\right) = 1-\alpha \;\Rightarrow t_{ji}^{\alpha} = F_{ji}^{-1}(1-\alpha) \qquad (2)$$

We also define a step function ($m_{ji}(\theta, td)$) which assigns weight to each request based on the remaining time to deadline (*td*). This function takes the value of one when the time to deadline is relatively large. As a request gets close to its deadline, this function takes the value of *M,* where *M* is a sufficiently large number. This gives a high priority to the service requests that are *critically* close to their deadline. Specifically, for a service request that is reaching its deadline in less than $\theta *$ *(1/μji),* this function returns a value of *M*. Finally for requests that already missed their deadlines the value of this function will be zero (see (3)).

$$m_{ji}(\theta, td) = \begin{cases} 1 & If\ td > \frac{\theta}{\mu_{ji}} \\ M & If\ 0 < td \le \frac{\theta}{\mu_{ji}} \\ 0 & If\ td = 0 \end{cases} \qquad (3)$$

The priority-based allocation index is calculated by multiplying function $m_{ji}(\theta, td)$ by the penalty cost (π_{is}) of violating the deadline and dividing it by t_{ji}^{α}. According to this index, service requests with a high penalty cost and an imminent deadline receive the highest priority while requests with a passed deadline are given the lowest priority. We note that there may be service requests for which the penalty function may be duration-based rather than a deadline-based, e.g. how long a web server remains down after an outage. This type of penalty may be associated with the most critical or revenue-generating components of IT infrastructure (e.g., a web server that processes payments). The present simulation model, however, does not address these types of penalty cost functions.

5.2 Simulation Results

In our initial testing, we consider a system consisting of four agents, four problem (service request) types as a way to incorporate their complexity, and three levels of priority. We assume that agents have different skill levels. Table 1 illustrates the mean service time of each agent for each service request type. The penalty of missing the deadline for priorities 1, 2, and 3 are 100, 80 and 30 respectively. The deadlines for priorities 1, 2 and 3 are 40, 60 and 85 minutes respectively. Our performance criterion is to maximize the long run average SLA violation penalty per unit time.

Table 1: Mean service time ($1/\mu_{ji}$) for each agent and each service request type. Infinity represents that the agent is not skilled in handling the type of request.

	Service Request Type (i)			
Agent (j)	1	2	3	4
1	10	10	∞	∞
2	∞	∞	∞	3
3	10	∞	15	∞
4	∞	∞	5	8

In order to demonstrate the performance of the allocation indices, we simulate this system under two settings. In the first setting, service requests are assigned to agents based on a first-come-first-serve (FCFS) policy. In the alternative setting, we use the priority-based allocation index defined in (1). We use common random numbers for both systems in order to reduce the variance of the performance difference. We then analyze the warm up period using Welch's method, presented by Law and Kelton (2000). After examining the output we conclude that 100 service requests provides a sufficient warm-up period. After deleting the data from the warm-up period we calculate the average difference between the cost of our proposed algorithm and that based on FCFS policy and construct 90% confidence intervals.

Our initial computations show that the proposed dispatching algorithm dramatically reduces the SLA violation penalty cost compared to a FCFS policy. However, in spite of reducing the penalty cost, the proposed algorithm increases the average queue length and average delay in the queue. This is primarily due to the fact that the service requests passed their deadline receive the lowest priority and therefore have to wait longer. We summarize the results (SLA violation penalty) of 40 replications in Table 2.

Table 2: Comparing the long run average SLA violation penalty per unit time of the proposed dispatching policy with FCFS.

	FCFS Policy	Proposed Priority-based Dispatching Policy	Difference
Mean Performance	37.33	7.28	30.04
90% Confidence Interval	(35.42,39.23)	(6.76,7.81)	(28.04,32.05)

As seen in Table 2, the proposed dispatching policy significantly reduces the SLA violation penalty cost since the 90% confidence interval on the difference does not include zero.

5.3 Sensitivity Analysis

In this section we investigate the effects of (1) increasing the aggregate arrival rate, (2) improving the skill levels of the agents and (3) decreasing the deadlines on the SLA violation penalty.

To investigate the sensitivity of the proposed algorithm to any possible change in arrival rate, we analyze the system where the aggregate arrival rate is increased from 0.5 to 4 arrivals per minute (see Figure 3). We observe that the penalty cost increases for both algorithms. However, the rate of increase in our proposed algorithm is much slower compared to FCFS as the aggregate arrival rate increases. In other words, prioritizing the service requests is more critical as the system becomes more crowded.

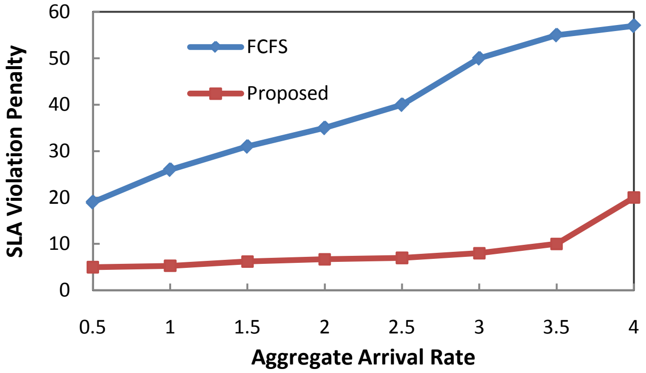

Figure 3: SLA violation penalty as a function of arrival rate

The second important analysis here is to show how the SLA violation cost changes as the agent's skill levels improve marginally. In this case we decrease the mean service time of one of an agent (Agent 1) for service request type two ($1/\mu_{12}$) from 10 to 8 minutes. There are many potential ways to achieve this objective in a real-life delivery environment, e.g. via customized training programs, streamlining the problem diagnosis process, or automating some of the tasks performed by the agent. The results (see Figure 4) show that the proposed algorithm takes advantage of this improvement; however, the improvement under FCFS is not significant. This result suggests that the overall performance of the system in terms of cost is highly sensitive to the efficiency of the dispatching system. Moreover, improving the skill levels of agents does not necessarily decrease the penalty cost associated with deadline violation, because the policy employed may not be effective.

Finally, we examine how our proposed algorithm responds to modifications of the terms of the service contract, particularly, shortening the deadlines. In order to test this scenario, we reduce the deadline for service requests of priority level 1, from 40 to 20 minutes. Our results (see Figure 4) show that both the proposed and FCFS policies are very sensitive to this reduction. In both settings the SLA violation penalty is increased drastically. However the increase in the total penalty cost is more pronounced under the FCFS policy. This result is consistent with our hypothesis that a sound dispatching policy can better dampen the negative impacts of environmental factors, in this case, the terms of the contract.

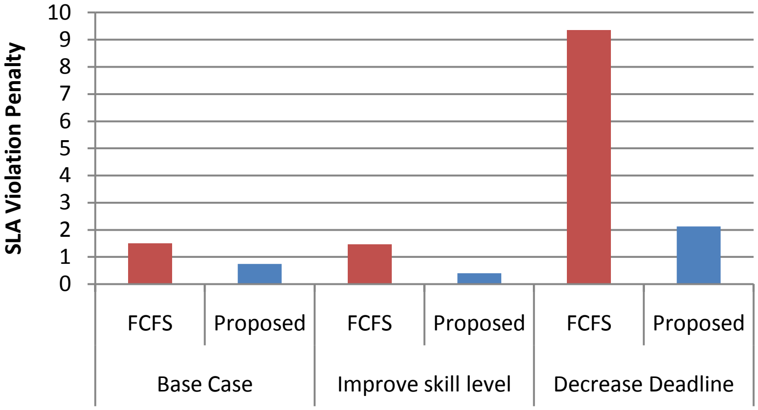

Figure 4: Effects of improving agents' skill levels and decreasing the deadlines on the overall cost of the system

6 CONCLUSION AND FUTURE CHALLENGES

In this research, we propose a new priority-based dispatching policy that incorporates the inherent complexity of the IT service delivery environment. In the proposed approach, we first introduce a more accurate service complexity categorization by refining the granularity level of our input model. We also develop a new allocation index that captures uncertainty in agents' service time, which is a significant source of variability commonly observed in such environments. This index considers important factors such as deadlines and the variability in service time. It also incorporates the nature of probability distributions to address a more accurate measure of variability.

Our proposed dispatching algorithm assigns a priority-based allocation index to each service request in the queue. This index is dynamically updated upon each service termination in the system (i.e., we assume non-preemptive service). Our initial results show that the proposed dispatching policy can have a significant impact on reducing the penalty cost of violating deadlines. Sensitivity analysis shows that the proposed dispatching policy is even more significant in reducing penalty cost when the aggregate arrival rate increases or deadlines are shortened (both cases represent a more congested system in some sense). Further benchmarking is warranted.

In this research, we restricted attention to only non-idling policies. However, there is no guaranty that our proposed policy can always perform well in other settings such as the case where idling is allowed. In this particular problem setting where agents are cross-trained, one may argue that keeping some skilled agents idle for short periods of time may produce better results. Consider this simplified example: Agent 1 is very efficient at resolving service requests of type 1. At a given time, Agent 1 becomes idle but there is no request of type 1 in the queue. Here we have two choices: we can either assign another service request (e.g. type 2 at which Agent 1 is not so skilled) or wait for a certain period of time expecting that another service request of type 1 arrives. In the current framework, we only restrict our approach to the first option. Future investigations are required to address this type of policies.

ACKNOWLEDGMENTS

The authors would like to thank Kyle Cresci for his help in developing the simulation code. They also thank Shervin AhmadBeygi for his feedback on the original idea of this research and providing references. The work of Parvin and Van Oyen was supported in part by NSF grant DMI- 0542063. The work of Parvin was also partially supported through STIET program funded by NSF IGERT grant No. 0114368 and 0654014.

REFERENCES

Bartholdi III, J. J. 1981. A guearanteed-accuracy round-off algorithm for cyclic scheduling and set covering. *Operations Research* 29(3):501-510.

Brusco, M. J., Johns, T. R., and Reed, J. H. 1998. Cross-utilization of a two-skilled workforce. *International Journal of Operations and Production Management* 18(6):555-564.

Feldman, Z., Mandelbaum, A., Massey, W. A., and Whitt, W. 2008. Staffing of time-varying queues to achieve time-stable performance. *Management Science* 54(2):324-338.

Law, A. M., and Kelton, D. 2000. *Simulation Modeling and Analysis* .3rd ed. New York: McGraw-Hill, Inc.

Ma, Q., and Steenkiste, P. 1998. Routing traffic with quality-of-service guarantees in integrated services networks. *NOSSDAV*.

Pyzde, T. 1995. Why normal distributions aren't [all that normal]. *Quality Engineering* 7(4):769-777.

Sisselman, M. E., and Whitt, W. 2007. Value-based routing and preference- based routing in customer contact centers. *Production and Operations Management* 16(3):277-291.

Van Oyen, M. P., Gel, E. G., and Hopp, W. J. 2001. Performance opportunity for workforce agility in collaborative and noncollaborative work systems. *IIE Transactions* 33(9):761-777.

Wallace, R. B., and Whitt, W. 2005. A staffing algorithm for call centers with skill-based routing. *Manufacturing and Service Operations Management* 7(4):246-294.

Whitt, W. 2006. Staffing a call center with uncertain arrival rate and absenteeism. *Production and Operations Management* 15(1):88-102.

AUTHOR BIOGRAPHIES

HODA PARVIN is a Ph.D. student in the Industrial and Operations Engineering Department at the University of Michigan. She received her Master of Science degree in Industrial and Systems Engineering from Texas A&M University. Her research focus, is currently on optimal assignment policies using Markov Decision Processes with applications in healthcare and service industries. She received a Rackham Merit Fellowship from the University of Michigan to support two years of study. In addition, she has received a STIET Fellowship to support another year of study and to provide coursework in economics, information, and mathematics. Her email is <hoda@umich.edu>

ABHIJIT BOSE is a Research Scientist with IBM T. J. Watson Research Center. He received his doctoral degrees in Computer Science and Engineering Mechanics from University of Michigan-Ann Arbor and University of Texas at Austin, respectively. His research focus is system modeling and optimization focusing on large-scale service systems and data center management systems. He currently leads a global team of researchers and engineers across IBM to design and develop next-generation quality management systems. His email address is <bosea@us.ibm.com>

MARK P. VAN OYEN is presently an Associate Professor of Industrial and Operations Engineering at the University of Michigan, where he also serves as the Director of the Engineering Global Leadership Honors Program for the College of Engineering. His core interests focus on the analysis, design, control, and management of operations systems, with emphasis on healthcare, service operations, and supply chains and how they can be designed for greater performance, flexibility, and resilience. His research also contributes to applied probability and the control and performance analysis of queueing networks. His awards include the 2009 IOE Dept. Faculty of the Year, ALCOA Manufacturing Systems Faculty Fellow, a best paper award from *IIE Transactions,* and Researcher of the Year from Loyola U. Chicago's School of Business. He has served as Associate Editor for *Operations Research, Naval Research Logistics*, and *IIE Transactions* and Senior Editor for *Flexible Services and Manufacturing.* He has served as a faculty member in Northwestern University's Sch. of Engr. (1993-2005) and Loyola University of Chicago's Sch. of Bus. Admin. (1999-2005). In industry, he worked on the research staff of GE Corporate R&D as well as in analysis and simulation for Lear Siegler's Instrument and Avionic Sys. Division. His email is <vanoyen@umich.edu>