Probability in the Engineering and Informational Sciences, **26**, 2012, 581–591.
doi:10.1017/S0269964812000216

APPROXIMATE DYNAMIC PROGRAMMING TECHNIQUES FOR SKILL-BASED ROUTING IN CALL CENTERS*

D. Roubos and S. Bhulai
VU University Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands
E-mails: d.roubos@vu.nl; s.bhulai@vu.nl

We consider the problem of dynamic multi-skill routing in call centers. Calls from different customer classes are offered to the call center according to a Poisson process. The agents are grouped into pools according to their heterogeneous skill sets that determine the calls that they can handle. Each pool of agents serves calls with independent exponentially distributed service times. Arriving calls that cannot be served directly are placed in a buffer that is dedicated to the customer class. We obtain nearly optimal dynamic routing policies that are scalable with the problem instance and can be computed online. The algorithm is based on approximate dynamic programming techniques. In particular, we perform one-step policy improvement using a polynomial approximation to relative value functions. We compare the performance of this method with decomposition techniques. Numerical experiments demonstrate that our method outperforms leading routing policies and has close to optimal performance.

1. INTRODUCTION

Call centers deliver a huge variety of services, and therefore continue to grow in size. From the perspective of economies of scale this is beneficial [9]. At the same time, it can be undesirable to have only fully cross-trained agents who can handle all service requests, due to high personnel costs and switchover times between different services.

*This study is an addendum to [3].

© Cambridge University Press 2012 0269-9648/12 $25.00

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press

Skill-based routing is the solution. Instead of having fully cross-trained agents who possess all skills, agents are grouped into heterogeneous pools with different skill sets. Upon arrival of a call, the automatic call distributor (ACD) has to decide to which agent group the call is assigned or it can decide to queue the call. Obviously, there are many policies to deal with this decision problem leading to a number of problems. First, finding the optimal policy for the ACD is a hard problem for large call centers [10], and second, the decision instructions should be efficient so that little time is necessary for making good decisions online.

The problem of assigning a call to an agent group upon arrival is also called the agent selection problem. Next to this decision problem, the ACD also has to select a possibly waiting call when an agent becomes available. This problem is called the call selection problem. The optimal policy for the agent and call selection problem is hard to derive. The state space of the problem limits many numerical methods to small-sized call centers [4,7,8,12]. Therefore, many good routing rules or heuristics have been proposed. In [15–17], the authors consider fixed, static priority policies. Overflow routing [10] and [5] provides an approximate analysis of the overflow behavior from one pool of agents to another. For a literature survey on asymptotic heavy-traffic regimes we refer to [11] and [6].

In this study we deal with the agent and call selection problem in the framework of Markov decision processes (MDPs). Owing to the large state space that the formulation brings forth standard algorithms are not applicable. To alleviate this problem, we use approximate dynamic programming (ADP) techniques to derive nearly optimal policies. In particular, we shall use one-step policy improvement to derive our dynamic routing policies. To this end, one has to select a reasonable initial policy and derive its relative value function. A reasonable initial policy is the overflow routing policy. In this setting it is natural to study each agent group separately, and to try to find an accurate value function or approximation for those decomposed smaller systems. However, the analysis is difficult, since the overflow process is not Markovian [14]. We solve this problem by approximating the value function for the overflow process.

The decomposition approach has been applied earlier in [3]. The author assumes that the overflow process is a Poisson process, with rates that are approximated using the fraction of time that an arbitrary arriving call finds all agents in an agent group occupied. This fraction is determined using the well-known Erlang-B formula, which is for blocking systems rather than queueing systems. The approximation in [3] performs well compared with the optimal policy. However, the results show that there is still room for improvement. We adopt a method that combines the value function approximation and simulation to estimate the parameters of the arrival processes to each of the decomposed systems resulting in better policies.

An alternative approach to solve the agent and call selection problem is to approximate the value function of the whole system directly, without using a decomposition approach. This eliminates the difficulty of estimating the parameters for each of the decomposed systems. We show the performance of both methods. More specifically, the study is organized as follows. In Section 2 we formulate the model of the call center. In Section 3 we introduce our solution method, which is followed by Section 4

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press

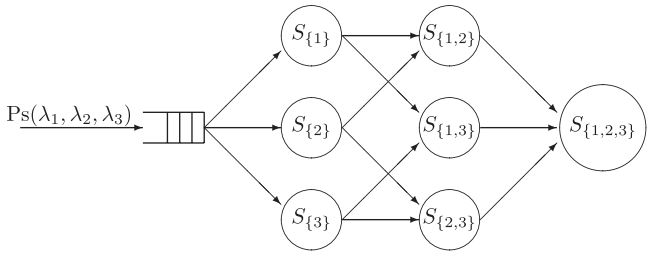


FIGURE 1. A call center representation with three types of skills.

in which we show numerical experiments. We conclude in Section 5 with conclusions and ideas for further research.

2. MODEL FORMULATION

We consider a multi-skill call center at which calls arrive and are served by an agent with the right skill. Denote by $\mathcal{S} = \{1, \ldots, N\}$ the set of skills, so that there are N different skills. Each agent can have any mixture of skills, denoting the types of calls that the agent can handle. Let $\mathcal{G} = \mathcal{P}(\mathcal{S})$ be the different agent groups. Note that $\mathcal{P}(\mathcal{S})$ is the power set of all skills. For three different skills, this would already imply seven different agent groups, as is depicted in Figure 1. We call a group $G \in \mathcal{G}$ with $|G| = 1$ (agents with a single skill) specialists, the group with $|G| = |\mathcal{S}| = N$ (agents with all skills) generalists, and groups with $1 < |G| < |\mathcal{S}|$ cross-trained agents. For notational convenience, let $\mathcal{G}^s = \{G \in \mathcal{G} \mid s \in G\}$ for $s \in \mathcal{S}$, that is, the agents groups that contain skill s. Calls of type $s \in \mathcal{S}$ arrive according to a Poisson process with rate λ_s. If there is no agent available who can handle the call of type s, it will be queued in a waiting buffer of infinite size. Each agent group $G \in \mathcal{G}$ consists of S_G agents and serve the calls with independent exponentially distributed amount of time with parameter μ_G. Upon a service completion, the agent can take a call out of the waiting buffer. Let $\mathcal{X}$ denote the state space. With $(\vec{q}, \vec{x}) \in \mathcal{X}$ we denote by q_i the number of calls in the queue that require skill $i \in \mathcal{S}$, and by x_G the number of occupied servers in group $G \in \mathcal{G}$. The objective is to minimize the number of calls in the system, that is, the number of calls that are being handled and the number of waiting calls. Therefore, let the cost–function be $c(\vec{q}, \vec{x}) = \sum_{i \in \mathcal{S}} q_i + \sum_{G \in \mathcal{G}} x_G$ for all $(\vec{q}, \vec{x}) \in \mathcal{X}$.

We want to determine the optimal assignment of calls to agents. We have two types of problems to solve: the agent selection problem and the call selection problem. For the agent selection problem, the possible actions are to assign the arriving call of type s to an available agent in one of the groups $\mathcal{G}^s$, or to put the call into the waiting buffer. Upon a service completion in group $G \in \mathcal{G}$, the possible actions for the call selection problem are to get a waiting call of one of the types group G can serve, or to leave the agent idle. Denote by Π all deterministic policies. A policy $\pi \in \Pi$ is a mapping from states to actions, describing which action to take in each state. Note that

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press

if there is no function available that easily describes this mapping, then only a look-up table is possible.

Let us consider the policy of overflow routing. More specifically, let an $N \times |\mathcal{G}|$ matrix describe the priority of call types to groups of agents. We form such a matrix by letting agent groups with smaller skill sets have priority over agent groups with larger skill sets. Therefore, for the agent selection problem, the specialists have the highest priority, and upon arrival of a call of type s, it is assigned to an available agent in the group of specialists who can handle type s calls. If none of the agents are available in that group, it is assigned to one of the agent groups consisting of two skills. If that is not possible either, than groups consisting of three skills are considered, and so on, until the group of generalists is reached. If none of the generalists are available, then the call is put in the waiting buffer. For the call selection problem we rank the priorities according to the call types, and give calls of a lower-type priority over calls of a higher type, meaning that call type i gets priority over call type j if $i < j$. We take these two heuristics as our initial policy π^0, and improve on it.

After uniformizing the system [13] and given our initial policy π^0, we obtain the Poisson equations given by

$$g + \left[\sum_{i\in\mathcal{S}} \lambda_i + \sum_{G\in\mathcal{G}} S_G \mu_G\right] V(\vec{q},\vec{x}) = \sum_{i\in\mathcal{S}} q_i + \sum_{G\in\mathcal{G}} x_G + \sum_{i\in\mathcal{S}} \lambda_i h_i(\vec{q}+e_i,\vec{x}) + \sum_{G\in\mathcal{G}} x_G \mu_G \tilde{h}_G(\vec{q},\vec{x}-e_G) + \sum_{G\in\mathcal{G}} (S_G - x_G)\mu_G V(\vec{q},\vec{x}), \quad \textbf{(1)}$$

where g is the average costs, $V(\vec{q},\vec{x})$ the relative value function, and $h_i(\vec{q},\vec{x})$ and $\tilde{h}_G(\vec{q},\vec{x})$ are the functions that correspond to the initial policy π^0 described by the priority routing matrix as aforementioned. For the case with $N = 3$ and $\mathcal{G} = (G_1, G_2, G_3, G_4) = (\{1\},\{2\},\{3\},\{1,2,3\})$ we have $h_i(\vec{q},\vec{x})$ and $\tilde{h}_G(\vec{q},\vec{x})$ given by

$$h_i(\vec{q},\vec{x}) = \begin{cases} V(\vec{q}-e_i,\vec{x}+e_i), & \text{if } q_i > 0 \text{ and } x_i < S_{G_i}, \\ V(\vec{q}-e_i,\vec{x}+e_4), & \text{if } q_i > 0,\ x_i = S_{G_i}, \text{ and } x_4 < S_{G_4}, \\ V(\vec{q},\vec{x}), & \text{otherwise} \end{cases}$$

and

$$\tilde{h}_G(\vec{q},\vec{x}) = \begin{cases} V(\vec{q}-e_i,\vec{x}+e_i), & \text{if } q_i > 0 \text{ and } G = \{1\},\{2\},\{3\}, \\ V(\vec{q}-e_i,\vec{x}+e_4), & \text{if } G = \{1,2,3\} \text{ and } i = \arg\min\{q_i | q_i > 0\}, \\ V(\vec{q},\vec{x}), & \text{otherwise.} \end{cases}$$

Note that $\mathcal{X}$ is countably infinite and very large making standard techniques such as value iteration inapplicable to derive the value function V. However, suppose that we have $V(\vec{q},\vec{s})$, or an approximation for V, available, then we could find a better policy

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press

by considering

$$\min\{V(\vec{q},\vec{x}),\{V(\vec{q}-e_i,\vec{x}+e_G)\mid i\in\mathcal{S},G\in\mathcal{G}^i,q_i>0,x_G<S_G\}\}. \quad (2)$$

At this point the importance of the relative value function V is definitely clear. The relative value function is needed to improve on our initial policy. Furthermore, using the value function V is a very efficient way to determine the action to take upon arrival of a call or upon a service completion of one of the agents. However, to obtain the value function analytically from (1) is very hard. Therefore, in the next section, we provide an approximation $\tilde{V}$ for the value function V by decomposing the system, followed by approximating the value function for each subsystem. Alternatively, we approximate the whole value function directly, without the decomposition approach.

3. METHOD

In this section, we study three different approximation methods for the call center problem formulation. The first approach is the one used in [3] and serves as a reference point to compare our approaches with. That approach assumes arrivals according to a Poisson process at each group of agents, and is referred to as Approach 1. The second approach is the method we proposed in the previous section, where we use a hyper-exponential distribution for the interarrival times instead of exponential. This method is referred to as Approach 2. The last approach, referred to as Approach 3, directly fits a parametrized function to the value function V in the Poisson equations (1). The methods are ranked in increasing order of computational time requirements and also in increasing quality of the policy. As such, the three methods display the wide range of possibilities for use in practice in call centers. In the sequel, we discuss each approach in more detail.

For the discussion of the methods, we focus on the case in which there are only specialists and generalists. Therefore, the set of agent groups $\mathcal{G}$ consists of the the sets $G_i=\{i\}$ (the specialists) and the set $G_{|\mathcal{S}|+1}=\{1,\ldots,N\}$ (generalists). The initial routing policy π^0 attempts to assign an arriving call of type i to the specialists G_i first. If all agents in group G_i are occupied, then one of generalists will be tried. If all generalists are also occupied, then the call is put in the waiting buffer of infinite size.

Approach 1: Suppose that we ignore the fact that calls can be put in a waiting buffer, and instead assume that calls are lost when they cannot be handled at the time of arrival. In that case, each of the agents groups can be seen as independent queueing systems with its own effective arrival process and service times. More specifically, for the groups $G_i, i=1,\ldots,N$ it is an $M/M/S/S$ queueing system, while for the group $G_{|\mathcal{S}|+1}$ it is a $G/M/S/S$ queueing system.

Let β be the expected service time, so that for exponentially distributed service times $\beta=1/\mu$. Furthermore, denote by $B(S,a)$ the steady-state probability that there are S calls in a $G/G/S/S$ system with offered load $a=\lambda\beta$. Then, the probability of overflow from agent group G_i to the generalists is equal to $B(S_{G_i},\lambda_i/\mu_{G_i})$, and

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press

therefore, the effective arrival rate to agent group G_i is given by $\lambda_i(1 - B(S_{G_i}, \lambda_i/\mu_{G_i}))$. For the group of generalists $G_{|\mathcal{S}|+1}$ the effective arrival process is difficult, since it is a mixture of different overflow processes, and each of them has hyperexponentially distributed interarrival times [14]. Instead of the hyperexponential distribution, this approach assumes arrivals according a Poisson process to the generalists with arrival rate equal to $\sum_{i\in\mathcal{S}} \lambda_i B(S_{G_i}, \lambda_i/\mu_{G_i})$.

Now that we have obtained the effective arrival rates, the assumption of lost calls is dropped since our call center system has a waiting buffer. This replaces the queueing systems $M/M/S/S$ by the $M/M/S$ system. Let $V_1(x, \lambda, \mu, S)$ be the relative value function for an $M/M/S$ queueing system. Then, by combining all parts, the approximation of the relative value function $\tilde{V}$ for the whole system, the call center, is given by

$$\begin{aligned}\tilde{V}(\vec{q}, \vec{x}) = &\sum_{i\in\mathcal{S}} V_1(q_i + x_i, \lambda_i(1 - B(S_{G_i}, \lambda_i/\mu_{G_i})), \mu_{G_i}, S_{G_i}) \\ &+ V_1((q_1 + x_1 - S_{G_1})^+ + \cdots + (q_N + x_N - S_N)^+ + x_{N+1}, \mu_{G_{N+1}}, S_{G_{N+1}}).\end{aligned}$$

The relative value function $V_1(x, \lambda, \mu, S)$ for an $M/M/S$ queueing system has already been studied in [2] for the general case. For our case, in which the system is subject to holding costs only, the value function is given by

$$V_1(x, \lambda, \mu, S) = \frac{g}{\lambda} \sum_{i=1}^{x} F(i) - \frac{1}{\lambda} \sum_{i=1}^{x} (i-1)F(i-1),$$

for $x = 0, \ldots, S$, and

$$\begin{aligned}V_1(x, \lambda, \mu, S) = &-\frac{(x-S)\rho}{1-\rho}\frac{g}{\lambda} + V_1(S) \\ &+ \left[\frac{(x-S)(x-S+1)\rho}{2(1-\rho)} + \frac{(x-S)(\rho + s(1-\rho))\rho}{(1-\rho)^2}\right]\frac{1}{\lambda},\end{aligned}$$

for $x = S, S+1, \ldots$, with $\rho = \lambda/(S\mu)$, and $F(x)$ and g defined by

$$F(x) = \sum_{k=0}^{x-1} \frac{(x-1)!}{(x-1-k)!}\left(\frac{\mu}{\lambda}\right)^k,$$

$$g = \frac{(S\rho)^S \rho}{S!(1-\rho)^2}\left[\sum_{n=0}^{S-1}\frac{(S\rho)^n}{n!} + \frac{(S\rho)^S}{S!(1-\rho)}\right]^{-1} + S\rho.$$

Approach 2: Contrary to Approach 1 that assumes Poisson arrivals everywhere, we take the interarrival times for the overflow process to be hyperexponentially distributed. Although the parameters for the hyperexponential distribution are hard to determine, we run a short simulation to obtain interarrival times and fit an H_2 distribution on these observations. Note that the estimated arrival rates to the specialists in Approach

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press

1 is a lower bound to the real arrival rate of these agent groups, because of the ignored waiting buffer. Therefore, we use simulation to obtain better estimates for their arrival rates as well. The fit of the H_2 distribution on the observations is done by using the expectation maximization (EM) method for phase-type distributions as is discussed in [1]. Hence, we model the queueing system for the generalists as an $H_2/M/S$ system.

Now that we have estimates for the effective arrival rates by means of simulation, we approximate the value function for the whole system by combining the relative values functions for the $M/M/S$ queues for the specialists and the $H_2/M/S$ queue for the generalists. Let again $V_1(x, \lambda, \mu, S)$ be the relative value function for an $M/M/S$ queueing system, and now let $V_2(x, P_2, \mu, S)$ be the relative value function for an $H_2/M/S$ queueing system. The approximation of the relative value function $\tilde{V}$ for the whole system, the call center, is given by

$$\tilde{V}(\vec{q}, \vec{x}) = \sum_{i \in \mathcal{S}} V_1(q_i + x_i, \lambda_i(1 - B(S_{G_i}, \lambda_i/\mu_{G_i})), \mu_{G_i}, S_{G_i}) + V_2((q_1 + x_1 - S_{G_1})^+ + \cdots + (q_N + x_N - S_N)^+ + x_{N+1}, P_2, \mu_{G_{N+1}}, S_{G_{N+1}}),$$

with P_2 the parameters for the H_2 distribution.

The relative value function $V_2(x, P_2, \mu, S)$ for an $H_2/M/S$ queueing system will be approximated by a parametrized function. To this end, consider a queueing system in which arrivals occur with interarrival times that are hyperexponentially distributed with two phases. With probability p and $(1 - p)$ the interarrival times are exponentially distributed with parameter λ_0 and λ_1, respectively. We say that the arrival process is in phase 0 and 1 if the next interarrival time has an exponential distribution with parameter λ_0 and λ_1, respectively. Denote by $(x, y) \in \mathcal{X} = \mathbb{N}_0 \times \{0, 1\}$ the state with x calls present in the system and that the arrival process is in phase y. Upon arrival of a call, the probability of the event that the next interarrival time is exponentially distributed with parameter λ_0 equals p. Now consider the Poisson equations for an $H_2/M/S$ queueing system, given by

$$\begin{aligned} V_2(x, y) + g = {} & x + (1 - y)\lambda_0(pV(x + 1, 0) + (1 - p)V(x + 1, 1)) \\ & + y\lambda_1(pV(x + 1, 0) + (1 - p)V(x + 1, 1)) \\ & + \min\{x, S\}\mu V([x - 1]^+, y) + (1 - (1 - y)\lambda_0 - y\lambda_1)V(x, y). \end{aligned}$$

We use the Bellman error minimization method to approximate $V_2(x, y)$ by an appropriate choice of the approximation structure. Let the approximation structure be given by

$$\tilde{V}_2(x, y, \vec{r}) = y \sum_{i=0}^{M} r_i x^i + (1 - y) \sum_{i=0}^{M} \tilde{r}_i x^i.$$

The rationale behind this is that we want to fit separate polynomial functions for each of the phases. To get a higher accuracy of the fit, even other types of approximation

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press

structures can be considered, for example, a function consisting of one polynomial for $x \leq S$, and a second polynomial for $x > S$.

Approach 3: The previous two approaches use decomposition to obtain a collection of subsystems for which the value function is known or can be approximated accurately. A drawback, however, is that the initial policy is not fully taken into consideration, for example, the priorities of different call types are not considered. To make better decisions based on the full initial policy, it is necessary to approximate the value function directly without a decomposition approach. Therefore, define a new state vector $y = (1, q_1 + x_1, \ldots, q_{|\mathcal{S}|} + x_{|\mathcal{S}|}, x_{|\mathcal{S}|+1}, \ldots, x_{|\mathcal{G}|})$. Given our initial policy π^0, we approximate the value function $V(\vec{q}, \vec{x})$ by

$$\tilde{V}(\vec{y}, \vec{r}) = \sum_{i=1}^{|y|} \sum_{j=i}^{|y|} r_{ij} y_i y_j. \tag{3}$$

Note that the structure is a polynomial approximation structure of degree 2, including cross-terms between different agent groups. The optimal parameter vector $\vec{r}$ can be obtained by minimizing the sum of squared Bellman errors or by approximate value iteration.

4. NUMERICAL EXPERIMENTS

In this section, we evaluate the performance of our ADP approaches for different parameter settings. First, we look at a call center with only specialists and generalists, and denote by x_7 and S_7 the number of calls at the generalists and the number of agents in the group of generalists, respectively. This notation corresponds to our definition of all agent groups, since the seventh group is the group consisting of agents who posses all skills. Second, we illustrate ADP for more complex call center systems, in which also partially cross-trained agents are present. In the first situation with only four agent groups, we can compute the optimal policy π^* by means of value iteration and we compare our ADP approaches to the optimal policy. However, for the second case, the computation of the optimal policy is computational intractable, and we can only look at the decrease in average cost compared to the initial policy π^0.

Now, consider the first situation with specialists and generalists. Let g, $g^{1'}$, $g^{2'}$, $g^{3'}$, and g^* denote the average cost corresponding to the initial policy, the one-step improved policies based on Approaches 1, 2, and 3, and the optimal policy, respectively. The percent relative error of the approximations with respect to the optimal policy is given by $\Delta^x = 100(g^{x'} - g^*)/g^*$, with $x = 1$ and 2 for both the decomposition approaches and $x = 3$ for the direct approximation. The results for different parameter settings are shown in Table 1.

The results in Table 1 show that the decomposition approaches improve upon the initial policy and give results that are close to optimal. The decomposition approach that uses hyperexponential interarrival times for the overflow process to the generalists

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press

TABLE 1. Numerical results for the situation with specialists and generalists

λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	μ_7	S_1	S_2	S_3	S_7	g^0	$\tilde{g}^1$	$\tilde{g}^2$	$\tilde{g}^3$	g^*	Δ^1	Δ^2	Δ^3
6	3	3	2	1	1	1	3	3	3	3	13.624	13.070	12.815	12.799	12.674	3.127	1.115	0.986
6	5	4	3	2	1	1	3	3	3	2	20.082	14.401	14.176	13.831	13.794	4.402	2.772	0.271
6	5	4	3	2	1	2	2	2	2	3	16.183	11.918	11.724	11.673	11.522	3.433	1.749	1.310
6	5	4	2	2	1	1	3	3	3	3	20.654	16.069	15.946	15.770	15.587	3.093	2.302	1.175
7	6	5	3	2	1	2	3	3	3	2	22.412	15.261	15.006	14.868	14.755	3.429	1.701	0.765
7	6	5	2	2	1	2	3	3	3	3	18.968	15.529	15.412	15.272	15.118	2.717	1.945	1.021

(Approach 2) performs better than the approach that assumes exponentially distributed interarrival times (Approach 1). This was to be expected, since clearly the process to the generalists is not exponential. For Approach 2, the obtained results are within 10% from the optimal cost after one step of policy improvement. However, the direct approximation approach (Approach 3) seems to perform much better with results that are within 3% from the optimal cost.

Next, we consider a more complex call center system with partially cross-trained agents as well. The calculation of the optimal policy using value iteration is very time consuming. However, to evaluate a fixed policy based on our approximations, we could also run a simulation. For the situation with seven agent groups, we only look at the performance of Approach 3, since that one turns out to be the most promising as we have seen in the situation with four agent groups. Here, we minimize the sum of squared Bellman errors to obtain the parameter vector $\vec{r}$. The set of representative states $\tilde{\mathcal{X}}$ is chosen randomly out of $\{0,\ldots,S_1+3\}\times\{0,\ldots,S_2+3\}\times\{0,\ldots,S_3+3\}\times\{0,\ldots,S_4\}\times\cdots\times\{0,\ldots,S_7\}$ with $|\tilde{\mathcal{X}}|=\alpha$. The weights $\omega(\vec{y})$ in Bellman error are chosen as $\omega(\vec{y})=\beta^{\sum_{i=1}^{|y|}y_i}$. Table 2 shows the results for a number of problem instances, where $\tilde{g}$ is the improved policy based on our approximation.

We have experimented with the choice of α and β. For each problem instance in Table 2 we used nine experiments and reported the best. The nine experiments correspond to the nine different possible combinations of α and β, with $\alpha=150, 250,$ and 350 and $\beta=0.5, 0.7,$ and 0.9. Given a policy obtained by our approximation, it is very easy and fast to evaluate the performance by means of simulation. Therefore, one can easily try different parameters in the Bellman error minimization method. We, however, observe that more states in the set of representative states and usually a slowly decaying weight function performs very well.

TABLE 2. Numerical results for the situation with cross-trained agents as well

λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	S_1	S_2	S_3	S_4	S_5	S_6	S_7	g^0	$\tilde{g}$	g^*	Δ
6	5	4	3	2	1	2	2	2	1	3	2	1	1	3	3	3	7.436	7.090	6.934	2.246
6	5	4	2	2	1	2	2	2	1	2	2	2	2	2	3	1	8.779	8.379	8.133	3.026
6	5	4	2	2	1	1	2	2	1	2	2	2	2	2	3	1	9.986	9.337	9.128	2.288
7	6	5	5	3	2	1	1	1	2	3	3	3	3	3	3	2	8.054	6.645	6.563	1.240
7	6	5	4	3	2	1	2	1	3	2	2	1	2	3	3	3	8.673	6.649	6.435	3.327

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press

5. CONCLUSIONS

In this study we have studied the dynamic skill-based routing problem in multi-skill call centers. Markov decision theory is a natural way to study dynamic state-dependent policies, while at the same time it suffers from the curse of dimensionality. Therefore, we have applied techniques from approximate dynamic programming to overcome this problem so that near-optimal policies can be obtained within reasonable time. Moreover, the method can be applied online and scales well with the size of the problem instance.

We have presented three different scalable approaches that differ in the computational time requirements and in the quality of the policies. The two least time-consuming methods are based on a decomposition technique. Although these two approaches already provide good policies and a significant improvement over static overflow policies that are used in practice, there is still room for improvement. The third method requires a bit more calculation time and is based on a second-order polynomial approximation (with all possible cross-terms) for the value function. This method outperforms the other approaches.

Further avenues of research could be to include abandonments to the call center. In principle, the proposed methods are general enough to compute the value function of a multi-server queue with abandonments. However, it requires more insight into the call center to choose an initial policy such that the relative value function can be approximated sufficiently well. This extension would also enable one to study mixed call center architectures, that is, queues with finite and infinite buffers, call types with finite and infinite patience, and so forth. It would be interesting to see how the model can be extended along these dimensions.

References

1. Asmussen, S., Nerman, O., & Olsson, M. (1996). Fitting phase type distributions via the em algorithm. *Scandinavian Journal of Statistics* 23(4): 419–441.
2. Bhulai, S. (2002). *Markov decision processes: The control of high-dimensional systems*. The Netherlands: Universal Press.
3. Bhulai, S. (2009). Dynamic routing policies for multi-skill call centers. *Probability in the Engineering and Informational Sciences* 23(1): 75–99.
4. Bhulai, S. & Koole, G.M. (2003). A queueing model for call blending in call centers. *IEEE Transactions on Automatic Control* 48: 1434–1438.
5. Franx, G.J., Koole, G., & Pot, S.A. (2006). Approximating multi-skill blocking systems by hyperexponential decomposition. *Performance Evaluation* 63: 799–824.
6. Gans, N., Koole, G., & Mandelbaum, A. (2003). Telephone call centers: tutorial, review, and research prospects. *Manufacturing Science and Operations Management* 5: 79–141.
7. Gans, N. & Zhou, Y. (2003). A call-routing problem with service-level constraints. *Operations Research* 51: 255–271.
8. Gurvich, I., Armony, M., & Mandelbaum, A. (2008). Service level differentation in call centers with fully flexible servers. *Mangement Science* 54: 279–294.
9. Koole, G. & Pot, A. (2006). An overview of routing and staffing algorithms in multi-skill cusomter contact centers. Technical report, VU University Amsterdam, March.

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press

10. Koole, G., Pot, A., & Talim, J. (2003). Routing heuristics for multi-skill call centers. In D. Ferrin, S. Chick, P.J. Sanchez, & D.J. Morrice (eds.), *Proceedings of the 2003 Winter Simulation Conference*. New Orleans, Lousiana.
11. Koole, G.M. & Mandelbaum, A. (2002). Queueing models of call centers: An introduction. *Annals of Operations Research* 113: 41–59.
12. Perry, M. & Nilsson, A. (1992). Performance modeling of automatic call distributors: Assignable grade of service staffing. In *XIV International Switching Symposium*. Yokohama, Japan, pp. 294–298.
13. Puterman, M.L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. New York: John Wiley & Sons, Inc.
14. Riordan, J. (1961). *Stochastic service systems*. New York: Wiley.
15. Shumsky, R.A. (2004). Approximation and analysis of a queueing system with flexible and specialized servers. *Operations Research Letters* 26.
16. Stanford, D.A. & Grassmann, W.K. (2000). Bilingual server call centres. In D.R. McDonald & S.R.E. Turner (eds.), *Call centres, traffic and performance*, vol. 28, pp. 31–48. Providence, RI: Fields Institute Communications.
17. Wallace, R. & Whitt, W. (2005). A staffing algorithm for call centers with skill-based routing. *Manufacturing Science and Operations Management* 7(4): 276–294.

https://doi.org/10.1017/S0269964812000216 Published online by Cambridge University Press