Research Report: Knowledge Management in Call Centers—How Routing Rules Influence Expertise and Service Quality

Geoff Ryder

Information Systems Management
and Computer Engineering Departments
Baskin School of Engineering,
University of California–Santa Cruz,
1156 High Street, Santa Cruz, CA 95064, USA
gryder@gmail.com

In a call center, customers are assigned to service agents by routing policies that seek to balance several objectives. Usually, these policies follow myopic rules in order to minimize the waiting time or maximize the quality experienced by the next customer. However, there is a secondary effect of the routing assignment: by learning-on-the-job, the development of the agents' expertise depends on the calls they take. In our research we seek to determine if and to what extent the myopic routing rules conflict with the goal of developing expertise—we seek to quantify the *earn-versus-learn* trade-off; and where myopic routing rules fail to perform well, we seek new ones that can guide contact center managers in their operational decisions.

This is a short report on new experiments to identify *earn-versus-learn* tradeoffs. We discuss a small system of two queues of different job types and three service agents, and use it to demonstrate candidate methods that may scale well for use in larger problems of the same type.

In our experiments, the first step is to solve an optimization problem whose objective function involves worker expertise, and whose constraints come from a queuing model that describes the system. The second step is to run a discrete event simulator to test routing policies that strive to implement the targets found by the optimization step.

---

1. Nonlinear Optimization Model

1.1 Modeling Expertise

Assume that a contact center's customer service agents gain expertise and–thereby improve their performance–by serving customers, and lose expertise through forgetting mechanisms in the absence of new customers to serve. Then over the long run, the agents' expertise levels will be determined by the average arrival rates of new customers.

We model the expertise of a service agent with simple dynamic equations, reflecting these essential features of gaining expertise through experience and lowering expertise through absence. We find that, in the long run, the expertise level of an agent increases as the arrival rate to this agent increases—a busy agent will maintain a higher level of expertise and therefore give the customers better average service. [1]

In a multi-agent environment, the arrival rates to each agent are influenced by the routing policy employed at the call center (Gans et al. 2003). Different routing policies may lead to different distributions of knowledge/expertise and therefore to different customer quality experiences.

Our approach accomodates the typical behavior of accepted experience curve models that match a range of empirical human performance measurements. Many such models of varying complexity have been proposed; see for example Badiru (1992), Shafer et al. (2001), Sikström and Jaber (2002), and Howick and Eden (2007). They are characterized by diminishing returns as the agent approaches the peak of his skill, and forgetting is commonly modeled as a negative power law or exponential process (Globerson and Levin 1987, Nembhard and Osothsilp 2001). We assume the quality of the service encounter increases if the worker retains more expertise (Pinker and Shumsky 2000, Whitt 2006).

Consider the evolution of expertise in an agent answering calls to a call center. Let the expertise $x(t)$ of the agent at time t be on a scale $0 \leq x(t) \leq 1$, where $x(t) = 0$ indicates a novice, and $x(t) = 1$ corresponds to an expert. Define the average time between completed jobs to be τ, including the receiving and processing of a job, followed by some time until the next job arrives. We assume that the agent learns while processing the job (on-the-job), thus increasing its expertise level x(t), and forgets while not processing, leading to a decrease of x(t).

In our learning model, the agent's expertise by processing one job increases on the average through

$$x(t) \mapsto x(t) + \alpha(1 - x(t))$$

where α is a learning parameter. That is, the experience gain is proportional to $(1 - x(t))$, and so becomes geometrically smaller as $x(t)$ approaches expert status. In the absence of forgetting, an agent will move from novice to 63% of the expert level by completing $1/\alpha$ jobs.

Skills need to be maintained through reinforcement; in the absence of work to occupy an agent, forgetting ultimately reduces the expertise of the agent to zero (novice level). We assume that forgetting occurs at a continuous rate β, so that for a period of length Δt, the expertise is discounted by $e^{-\beta \Delta t}$. Taking learning events and continuous forgetting together, we get

$$x(t + \tau) = (x(t) + \alpha(1 - x(t))e^{-\beta\tau}$$

Given these simple dynamics, asymptotic behavior of $x(t)$ will tend toward the fixed point x of this equation, with $0 \leq x \leq 1$. The smaller τ (i.e. the more jobs per time unit the agent is handling), the higher the asymptotic expertise level x, and vice versa.

---

[1] The motivation for the model was developed jointly with Professors Kevin Ross, Vijay Mehrotra, and Christoph Heitz. Prof. Heitz designed the asymptotic expertise equation.

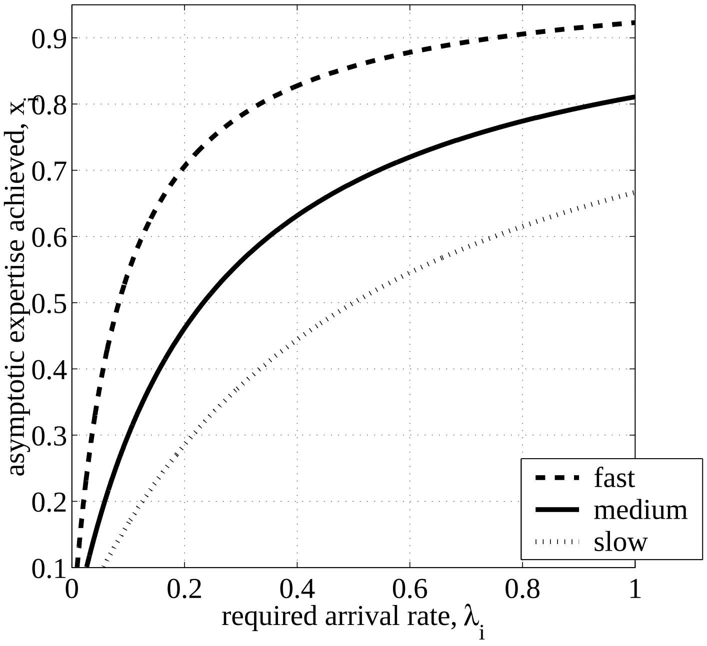

Figure 1: Plots of $x(\lambda)$ versus the steady state arrival rate λ. These are three different expertise functions x, Equation (2), parameterized with different values of α (learning) and β (forgetting). From the plot, we can see how large λ must be to support a specific expertise level, given α and β.

To model the expertise of agents in a contact center, let $x(t, \alpha_j, \beta_j)$ be the current state of an agent's expertise. Here t represents time, α_j a learning rate, and β_j a forgetting rate. Let τ_j the average interarrival time between jobs of type j seen by this agent. The evolution of expertise for job type j from time t to time $(t + \tau_j)$ can be modeled as follows.

$$x(t + \tau_j, \alpha_j, \beta_j) = (x(t) + \alpha_j(1 - x(t)))e^{-\beta_j \tau_j} \tag{1}$$

From this we can derive the fixed point value.

$$x_\infty = \frac{\alpha_j}{e^{\beta_j \tau_j} + \alpha_j - 1} \tag{2}$$

We can use the dynamic or the static asymptotic expertise value, $x(t)$ or x_∞, as an input parameter to functions that represent agent performance metrics, and observe how different levels of expertise affect customer service operations. For example, we may represent the probability of service failure $p_f(x)$ per call as a function of expertise; and in advanced models (with more complex constraints) we may consider the mean service rate $\mu(x)$ to depend on it as well. In this report we will focus on the expertise level itself, and less on how expertise manifests itself in specific performance metrics.[2]

From now on we will drop the subscript so that $x = x_\infty$; unless otherwise indicated x represents an asymptotic value given by Equation (2). In practice x will be reached after an

---

[2] This aspect is taken up in other ongoing work with Professor Mehrotra.

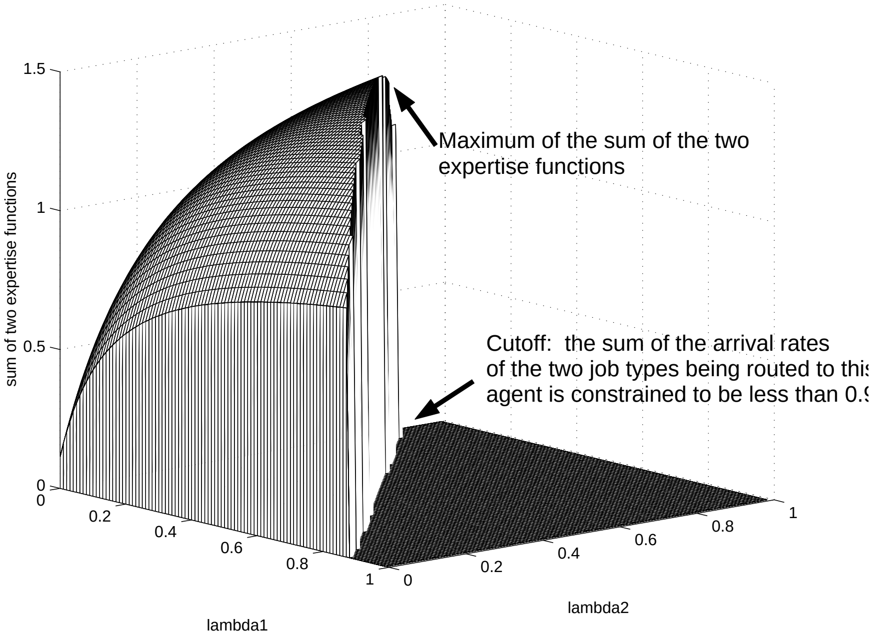

Figure 2: Plot of an agent's total asymptotic expertise. Consider an agent assigned to handle two job types. The 3D surface gives his total expertise $x_1(\lambda_1) + x_2(\lambda_2)$ as a function of his assigned steady-state arrival rate for job type 1, λ_1, and for job type 2, λ_2. His total utilization is constrained to be less than $\rho = (\lambda_1/\mu_1) + (\lambda_2/\mu_2) \leq 0.9$; and since $\mu_1 = \mu_2 = 1$ in this example, that forces the sum of allowed arrival rates to also be $\lambda_1 + \lambda_2 \leq 0.9$. The sum of the concave expertise functions is also concave, with the maximum joint expertise level at $(\lambda_1 = 0.45, \lambda_2 = 0.45)$.

agent processes a stream of a few hundred to a few thousand customer inquiries that reach her with non-bursty, consistently spaced arrival times.

Figure 1 shows three plots of asymptotic expertise functions x with different values of learning parameters. From the plot, we can see how large λ must be to support a specific expertise level, given α and β. The curve labeled "fast" has $\alpha = 0.0006$, and $\beta = 0.00005$; "medium" has $\alpha = 0.0006$, $\beta = 0.00014$; and "slow" has $\alpha = 0.0006$, $\beta = 0.0003$.

1.2 The Optimal Level of Expertise

From this discussion we see that a sufficient arrival rate $\lambda_{ij} = 1/\tau_{ij}$ is required for an agent to achieve and maintain a specific level of expertise. Now, the contact center modeled as a queueing system will have a set of arrival rates $\{\lambda_j\}$. Agents have learning and forgetting rates α_j and β_j that are strongly influenced by the type of job; they may also vary by individual agent i, giving α_{ij} and β_{ij}. In this work, we seek to solve a two-step decision problem defined as follows:

(i) Decide upon an assignment of a steady-state fraction of the total arrivals $\lambda_{ij} = \gamma \cdot \lambda_j$,

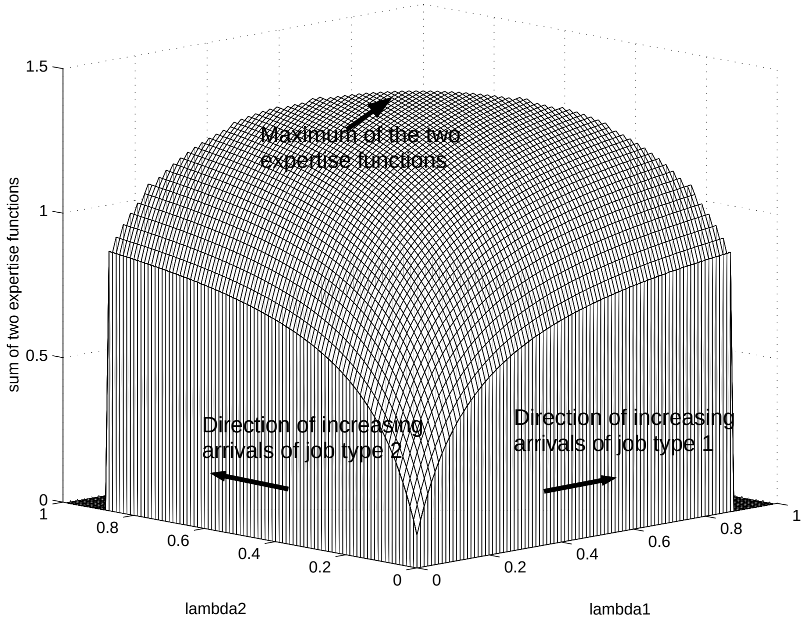

Figure 3: As Figure 2, one agent's total expertise $x_1(\lambda_1) + x_2(\lambda_2)$, rotated 90°.

$0 \le \gamma \le 1$, of each job type or queue j to each agent i, in such a way that the set of expertise levels among the workers in the call center is optimal. Denote that optimal set by $\{x^*_{ij}\}$. The meaning of "optimal" here may change depending on management's cost and performance targets.

(ii) Determine queue-to-agent routing rules for the multiple-queue, multiple-agent queueing system such that the optimal expertise levels determined in step (ii) are met.

A solver that handles nonlinear objective functions Z and linear constraints $(\mathbf{A} \cdot \vec{x} \ \le \ \vec{b})$ can compute the answer to the step (i) problem. How the optimization program is written will depend on the goal. As an example, suppose we wish to simply maximize the expertise of all the agents, where all customer types are considered equally valuable. The following program achieves this:

$$\boxed{\begin{array}{rrcll}
\max\limits_{\lambda_{ij}} & Z_1 & = & \sum\limits_{i=1}^{N_a} \sum\limits_{j=1}^{N_j} x_{ij}(\lambda_{ij}) & (3) \\
\text{such that} & \mathbf{T} \cdot \vec{\lambda_i} & \le & \vec{\rho_i} & (4) \\
 & \sum\limits_{i=1}^{N_a} \lambda_{ij} & = & \vec{\lambda_j} & (5) \\
\forall \text{ i, j:} & \lambda_{ij} & > & 0. & (6)
\end{array}}$$

Here we let N_a be the number of agents in the system, and N_j be the number of job types. Equation (3) maximizes the sum of every agent i's asymptotic or steady state expertise function $x_{ij}(\lambda_{ij})$, from Equation (2) for every job type j. We do this subject to constraints determined by the queueing system model. Inequality (4) constrains the sum of terms λ_{ij}/μ_{ij}, or mean steady state arrival rates divided by mean steady state service rates, to be less than a percentage utilization rate for every agent. Matrix $\mathbf{T}$ contains the values $T_{ij} = 1/\mu_{ij}$. Equation (5) forces all the arrivals to be served by someone. Inequality (6) forces all the arrival rate assignments to be positive, and the strictly-greater-than condition is needed to avoid a divide-by-zero error in Equation (2). In simulations λ_{ij} may be effectively set to zero.

Other objective functions may be used in place of Equation 3 to achieve different targets, while fulfilling the basic system requirements from Constraints (4), (5), and (6). For example:

$$\boxed{\max_{\lambda_{ij}} \quad Z_2 \quad = \quad \sum_{i=1}^{N_a} \sum_{j=1}^{N_j} \lambda_{ij} \cdot x_{ij}(\lambda_{ij}). \quad (7)}$$

Equation (7) is the same as Equation (3), but the expertise levels are weighted by the arrival rates λ_{ij}. This function maximizes the expertise seen by the average customer who arrives at the system. Another kind of objective would add a penalty for the service failure rate $p_f(x_{ij})$, which is a function of expertise, and hence also a function of λ_{ij}.

To help remember what these two objectives are, let us name Equation (3) *the firm's objective function*, or Z_1, and Equation (7) *the customer's objective function*, or Z_2. The firm's objective tries to create the most even distribution of expertise possible over all agents in the contact center. This benefits the firm by avoiding over-specialization in work assignments, thus providing a hedge against key specialized agents quitting, or against sudden unpredicted changes in demand λ_j. It also avoids variation in service quality, which helps in setting and meeting customer expectations.

The customer's objective tries to create a distribution of expertise that gives the highest possible *average* experience level seen by each customer. Of course a high average may obtained for a mix of highly experienced and very inexperienced workers, so this objective may lead to higher variation in service quality.

2. Routing Rule Simulation Model

In the previous section we described how to predict the optimal target values for the arrival rate to each agent for each job type, $\{\lambda^*_{ij}\}$, and the corresponding asymptotic expertise levels, $\{x^*_{ij}\}$. Yet these are static values that assume a long-run steady-state equilibrium – what is missing is how best to guide the evolution of the system over time to achieve and maintain these targets. In our queueing model of the system, that guidance will be given by *routing rules.* Such a rule will map the static target vector $\{\lambda^*_{ij}\}$, and the dynamically changing state of the queues and the agents' knowledge levels, to a decision about who takes the next job.

The discrete event simulation program shown in Figure 4 computes the state of the system over time, and shows how our different routing rules perform under different parameter

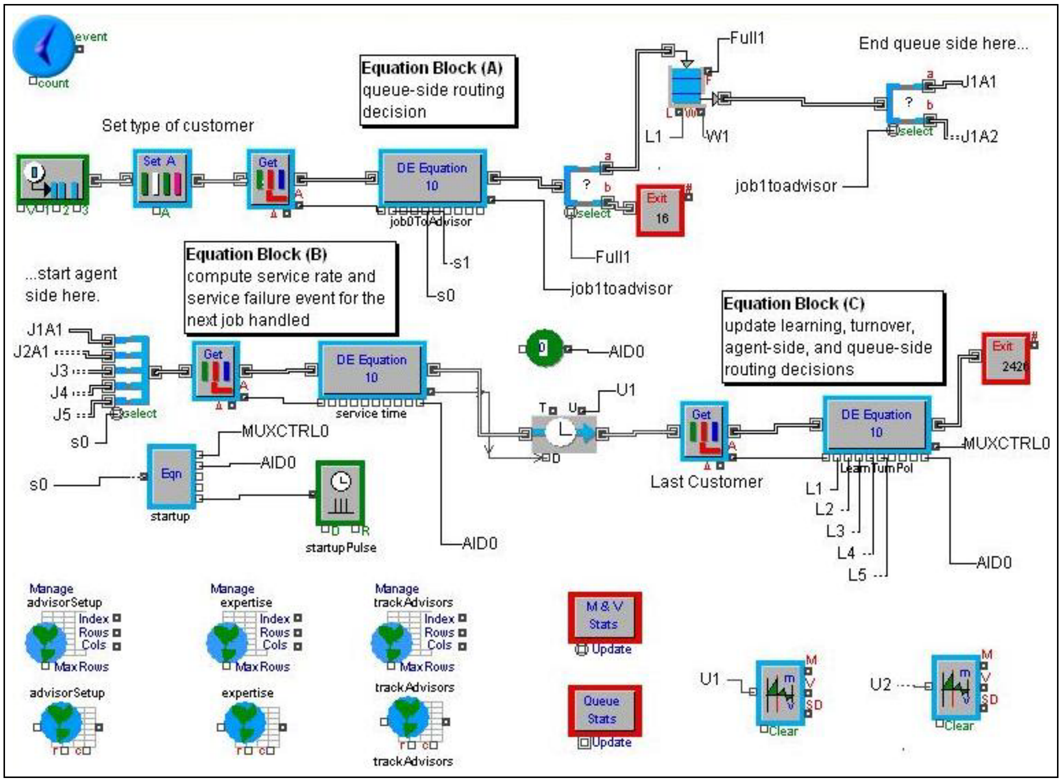

Figure 4: Discrete event simulation model.

settings. [3] In the figure, new customers arrive at the start of the chain of blocks near the top left corner, under the clock symbol. As they pass through each function block, that block is triggered to act and compute certain outcomes. Most of the work is done in the three programmable blocks named "DE Equation 10," which are described as follows:

(A) Customers with a specific type of issue to be resolved, or job type j, arrive at the system with interarrival times modeled by a negative exponential random variable with mean T_j. They either enter service directly, enter a queue specific to the job type, or are turned away if that queue j is too full. Equation Block (A) and the surrounding support functions handle the queue-side routing of customers: customers are randomly assigned among agents who are ready to accept a job of this type. It may happen that at one time, two or more agents wish to take customers from this queue to keep up with their target schedules. Block (A) ensures that the routing recipient is chosen fairly.

(B) Equation Block (B) determines the performance level of this agent for the current job type. That allows it to compute the next service time, as an exponential random variable with mean μ_{ij}; and whether or not a service failure occurs–the probability of the call going unresolved is a binomial random variable with $p = 0.1 \cdot (1 - x_{ij})$.

(C) Equation Block (C) updates the expertise level x_{ij} given the learning dynamics for this

---

[3] This is a Modl™ program in the Extend™ environment, Ver. 6, from ImagineThat! ® Inc.

agent and job type. It then observes the state of the queues and this agent's schedule, and implements a routing rule that determines which job type to request next.

Note that due to space limitations, Figure 4 shows only one job type and one agent; the queues and agents are replicated as needed to populate a full simulation. Customers are routed to agents from queues through the demultiplexer/multiplexer pairs, which are switched according to our routing rules. Various routing rules are possible for use in Block (C), such as:

- All agents swarm the longest queue, regardless of the state of their expertise: this is the "longest-queue-first" rule, or LQF.
- All agents strive to meet their individual optimal targets as given by $\{\lambda^*_{ij}\}$. We continuously estimate each proportion of arrivals $\hat{\lambda}_{ij}(t)$ from the number of jobs actually completed so far. If the agents are ahead of schedule and the online estimate gives $\hat{\lambda}_{ij}(t) > \lambda^*_{ij}$, keep serving customers, but favor the job type for which $\hat{\lambda}_{ij}(t) - \lambda^*_{ij}$ is smallest. This is the "target-no-idle rule," or TNI.
- All agents try to meet their optimal targets as in TNI, but when they are caught up, and $\hat{\lambda}_{ij}(t) = \lambda^*_{ij}$, they stop serving customers until new arrivals put them behind schedule again. This is the target-with-idle rule, or TWI.

We expect the TNI rule to perform well for heavily loaded systems, because the introduction of idling will cause queue lengths and customer waiting times to rise. We expect the TWI rule to perform well for large systems, because forced idling prevents chance from favoring some agents and allowing them to serve customers that should have gone towards developing other agents' expertise. The LQF rule is a good baseline case, because it is commonly used in contemporary contact centers; we would like to know under what circumstances other policies depart from LQF and whether they improve upon it by explicitly taking expertise development into account.

3. Simulation Examples

Three experiments suggest how our joint optimization of expertise targets and routing rules can identify useful trade-offs for contact center managers.

- Simulation 1 (LQF) tests the longest-queue-first routing rule. This rule ignores target λ assignments.
- Simulation 2 (TNI/Z1) tests the target-no-idle routing rule, with targets set by the optimization solver using the firm's objective function.
- Simulation 3 (TNI/Z2) uses target-no-idle and the customer's objective function.

Detailed descriptions of the input parameters and results are given at the end of the paper in Tables 2, 3, and 4. To summarize briefly here, there are two queues or job types $j \in \{1, 2\}$ in these simulations, and three agents $i \in \{1, 2, 3\}$. Arrivals are Markovian with rates $\lambda_1 = 0.6$

Property	Trial 1, LQF	Trial 2, TNI/Z1	Trial 3, TNI/Z2
Mean x_j, $j = 1$	0.317	0.322	**0.355**
Mean x_j, $j = 2$	0.236	0.251	**0.287**
Var x_j, $j = 1$	**1.58e-5**	6.77e-4	0.00739
Var x_j, $j = 2$	**0.000371**	0.002181	0.0144
Mean W_j, minutes, $j = 1$	2.49	7.39	**2.10**
Mean W_j, minutes, $j = 2$	2.54	5.11	**1.92**
Var W_j, minutes, $j = 1$	**8.61**	54.28	9.11
Var W_j, minutes, $j = 2$	14.01	41.11	**9.72**

Table 1: Summary statistics for all three experiments from Section 3, separately for job types $j \in \{1, 2\}$ and summed over all three agents. The best performance results appear in a boldface font. Here x_j represents expertise, and W_j the waiting time in a queue.

per minute and $\lambda_2 = 0.4$ per minute. The learning and forgetting dynamic that governs the evolution of expertise for each x_{ij} comes from one of the plots of Figure 1: *fast, medium,* or *slow.* All agents are *fast* for $j = 1$; but agent 1 develops at a *fast* rate for job 2, agent 2 develops at a *medium* rate for job 2, and agent 3 develops at a *slow* rate for job 2.

Figure 5 shows the *average expertise* seen by both types of customers under the different routing rules. For job type 1, where all agents are equally capable, TNI/Z2 performs about 10% better than LQF, and TNI/Z1 performs about 2% better than LQF. For job type 2, where the agents differ in capability, TNI/Z2 performs about 22% better than LQF, and TNI/Z1 performs about 6% better than LQF. This shows that actively seeking to optimize expertise levels may improve customer experiences on the average.

Yet this improvement comes at a price. Table 1 gives the performance on four metrics: mean expertise x (same as Figure 5), the variance of x experienced by the customer, the average time W spent by customers waiting in a queue, and the variance of W. It becomes clear that rule TNI/Z2 improved mean expertise at the cost of raising the variance; customers are treated less consistently. Rule TNI/Z1 is more consistent than TNI/Z2 with respect to the variance of expertise, but performs poorly on the waiting time metric. The target-with-idle rule (TWI) we described also perfoms very poorly on the waiting time metric – TWI is not a good candidate rule for use in a small system like this with just a few servers. Forcing servers to be idle is too costly in small systems.

4. Research Plan

To summarize, we have investigated the interplay of expertise targets and routing rules for contact centers. Our first step seeks the asymptotically optimal, steady-state distribution of expertise among the agents that is best suited to handle forecasted traffic over the long term. Then we seek routing rules that steer traffic to agents in order to meet those targets

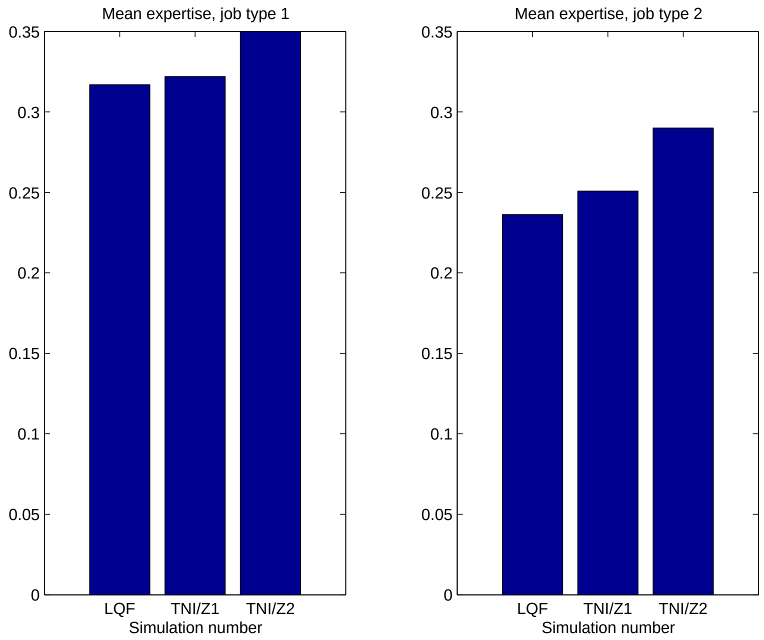

Figure 5: Left: average expertise level seen by customers for job type 1 for the three simulations discussed in Section 3. Right: the same for job type 2.

without compromising other performance metrics, such as customer waiting time.

For the small example in this report, we showed it was possible to improve the average customer experience at the cost of adding more variation, or inconsistency, to the process. This trade-off may be worth making, depending on the dimensions of customer service that improve with expertise. Those dimensions depend highly on the context. Contact centers for a hospital network, a credit card company, and an in-house IT department would view that trade-off differently.

We would like to improve on this basic model of knowledge distribution among agents. An important difference between this model and real contact centers is the doubly stochastic nature of real demand fluctuation: the properties of the arrival distribution will change with time. The asymptotic expertise targets we describe may not be accurate if the demand is "bursty." Another factor preventing us from reaching the expertise targets is employee turnover, which can be 25% annually in some contact centers. Accomodating bursty demand, turnover explicit waiting time, and other factors in the target-setting objective function is an important and practically inexhaustible line of research.

Regarding transient cases and routing rules, a large literature exists on multi-armed bandit problems and index routing rules for queueing systems–see Bertsekas (1995), for example–that may provide insights on the best way to route jobs to agents to achieve our expertise targets. As we scale up to investigate larger systems, we expect to see a diverse set of rules running at the same time in one facility to govern the work of different agent subgroups. Some will promote balanced expertise, like LQF, and some will promote specialization, as TNI/Z2 did for the second agent. Individual agents may also be subject to different routing rules at once that cover different segments of their overall work assignments.

Acknowledgements

The author would like to thank and ImagineThat! ® Incorporated for their generous support of our research, and Professors Kevin Ross, Vijay Mehrotra, Christoph Heitz, and John Musacchio for their invaluable insights.

References

Badiru, A. 1992. Computational survey of univariate and multivariate learning curve models. *IEEE Transactions on Engineering Management* **39**(2) 176–187.

Bertsekas, D.P. (1995) *Dynamic Programming and Optimal Control*, Vol. 2, Athena Scientific, Belmont, Massachusetts, USA, pp.242–256.

Gans, N., Koole, G., and Mandelbaum, A. 2003. Telephone call centers: tutorial, review, and research prospects. *Manufacturing and Service Operations Management* **5**(2) 79–141.

Globerson, S., Levin, N. 1987. Incorporating forgetting into learning curves. *International Journal of Operations and Production Management* **7**(4) 80–94.

Howick, S. and Eden, C. 2007. Learning in disrupted projects: on the nature of corporate and personal learning. *International Journal of Production Research* **45**(12) 2775–2797.

Nembhard, D.A., and Osothsilp, N. 2001. An empirical comparison of forgetting models. *IEEE Transactions on Engineering Management* **48**(3) 283–291.

Pinker, E., and Shumsky, R. 2000. The efficiency-quality trade-off of cross-trained workers. *Manufacturing and Service Operations Management* **2**(1) 32–48.

Shafer, S., Nembhard, D., and Uzumeri, M. 2001. The effects of worker learning, forgetting, and heterogeneity on assembly line productivity. *Management Science* **47**(12) 1639–1653.

Sikström, S., and Jaber, M. 2002. The power integration diffusion model for production breaks. *Journal of Experimental Psychology: Applied* **8**(2) 118–126.

Whitt, W. 2006. The impact of increased employee retention on performance in a customer contact center, *Manufacturing and Service Operations Management* **8**(3) 235–252.

Input Parameters, Simulation 1: Expertise Targets Ignored, Routing Rule LQF							
Symbol	Description	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3
j	job type	type 1	type 1	type 1	type 2	type 2	type 2
α	learning rate	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006
β	forgetting rate	0.00014	0.00014	0.00014	0.00005	0.00014	0.0003
λ_{ij}^*	target λ assignment	0.33	0.34	0.34	0.33	0.34	0.33
μ_{i1}, μ_{i2}	service rate	1	1	1	1	1	1
λ_{j1}	total arrival rate, type 1 jobs	0.6 jobs per minute					
λ_{j2}	total arrival rate, type 2 jobs	0.4 jobs per minute					
–	total simulation time	12000 minutes					

Simulation Results		
u_i	utilization for each agent, out of 100%	$u_1 = 32.5\%$, $u_2 = 34.6\%$, $u_3 = 33.0\%$
W_1	waiting time in queue 1	mean 2.49 minutes, variance 8.61 minutes
W_2	waiting time in queue 2	mean 2.54 minutes, variance 14.04 minutes
$\bar{x}$	expertise, job type 1	mean 0.3169, variance 1.58e-5
$\bar{x}$	expertise, job type 2	mean 0.2363, variance 3.71e-4

Table 2: Input parameters for Simulation 1, which tests the routing rule ”longest-queue-first” (LQF). This rule ignores target λ assignments.

Input Parameters, Simulation 2: Firm’s Objective (Z_1), Routing Rule TNI							
Symbol	Description	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3
j	job type	type 1	type 1	type 1	type 2	type 2	type 2
α	learning rate	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006
β	forgetting rate	0.00014	0.00014	0.00014	0.00005	0.00014	0.0003
λ_{ij}^*	target λ assignment	0.33	0.34	0.34	0.469	0.239	0.292
μ_{i1}, μ_{i2}	service rate	1	1	1	1	1	1
λ_{j1}	total arrival rate, type 1 jobs	0.6 jobs per minute					
λ_{j2}	total arrival rate, type 2 jobs	0.4 jobs per minute					
–	total simulation time	12000 minutes					

Simulation Results		
u_i	utilization for each agent, out of 100%	$u_1 = 35.2\%$, $u_2 = 31.4\%$, $u_3 = 33.7\%$
W_1	waiting time in queue 1	mean 7.39 minutes, variance 54.28 minutes
W_2	waiting time in queue 2	mean 5.105 minutes, variance 41.107 minutes
$\bar{x}$	expertise seen by customers, job type 1	mean 0.322, variance 6.77e-4
$\bar{x}$	expertise seen by customers, job type 2	mean 0.2508, variance 2.20e-3

Table 3: Input parameters for Simulation 2, which tests the routing rule ”target-no-idle” (TNI). This rule implements target λ assignments. The targets are set using the firm’s objective function, Equation (3).

Input Parameters, Simulation 3: Customer's Objective (Z_2), Routing Rule TNI							
Symbol	Description	Agent 1	Agent 2	Agent 3	Agent 1	Agent 2	Agent 3
j	job type	type 1	type 1	type 1	type 2	type 2	type 2
α	learning rate	0.0006	0.0006	0.0006	0.0006	0.0006	0.0006
β	forgetting rate	0.00014	0.00014	0.00014	0.00005	0.00014	0.0003
λ^*_{ij}	target λ assignment	0.334	0.332	0.334	0.025	0.95	0.025
μ_{i1}, μ_{i2}	service rate	1	1	1	1	1	1
λ_{j1}	total arrival rate, type 1 jobs	0.6 jobs per minute					
λ_{j2}	total arrival rate, type 2 jobs	0.4 jobs per minute					
–	total simulation time	12000 minutes					

Simulation Results		
u_i	utilization for each agent, out of 100%	$u_1 = 35.2\%$, $u_2 = 31.4\%$, $u_3 = 33.7\%$
W_1	waiting time in queue 1	mean 2.10 minutes, variance 9.11 minutes
W_2	waiting time in queue 2	mean 1.92 minutes, variance 9.72 minutes
$\bar{x}$	expertise seen by customers, job type 1	mean 0.35, variance 7.39e-3
$\bar{x}$	expertise seen by customers, job type 2	mean 0.28, variance 0.0144

Table 4: Input parameters for Simulation 3, which also tests the routing rule "target-no-idle" (TNI) using the customer's objective function, Equation (7).