Heuristics in Rostering for Call Centres

Shane G. Henderson, Andrew J. Mason
Department of Engineering Science
University of Auckland
Auckland, New Zealand
sg.henderson@auckland.ac.nz, a.mason@auckland.ac.nz

Ilze Ziedins
Department of Statistics
University of Auckland
Auckland, New Zealand
ziedins@stat.auckland.ac.nz

Richard Thomson, Tim Seabrook, David Burgess
Voice Technology Ltd
Auckland, New Zealand
richardt@voicetech.co.nz

Abstract

An important new feature on the business scene is the development of call centres, whereby a pool of staff is used to answer incoming calls from customers. This project develops a model that enables staffing levels to be determined to meet specified quality targets on customer wait times. Unlike most previous work, this new model explicitly considers call arrival rates that vary during the day, and exploits linkage between periods to keep staffing costs to a minimum.

1 Introduction

Many companies now employ a call centre, whereby a pool of staff is used to answer incoming calls from customers. By far, the largest component of cost in call centre operation is staffing cost. Therefore, it is highly desirable to keep staffing levels to a minimum throughout the day. Of course, the objective of minimizing staffing costs is subject to the constraint that customer service remain at a reasonable level throughout the day. We refer to the problem of minimizing staffing (rostering) costs subject to maintaining a minimum level of customer service as "the rostering problem". The problem is nontrivial because demand typically varies considerably throughout the day, and there is usually great flexibility in staffing decisions. This paper discusses work performed as part of a Masters in Engineering Thesis [8] by Richard Thomson for VoiceTech Systems under the guidance of the other authors.

Perhaps the primary measure of customer service is customer waiting time in the queue before reaching a server. It is also desirable to incorporate customer abandonment (reneging), where a customer hangs up before receiving service, in any measure of customer service. This is usually achieved by defining a customer specific Customer Grade of Service (CGOS). The CGOS is typically expressed as a percentage, with 100% reflecting ideal service, and lower percentages representing lower levels of service. For example, a customer that abandons might receive a CGOS of 0%. The constraint that customer service be satisfactory may be expressed in many different forms. For example:

- CGOS should exceed 50% for all customers.
- 95% of customers should receive a CGOS > 80%.
- In any 2 hour window, the average CGOS should exceed 80%.
- During peak times, the average CGOS should exceed 50%, while at other times it should exceed 80%.
- The expected CGOS for a customer arriving at any time throughout the day should exceed 80%.

Because of the random nature of call centre operation, for any given day, these requirements (with the possible exception of the last one) can only be met with a certain probability. It is usual to perform some form of averaging of CGOS to obtain an overal grade of service (GOS) for a given set of customers.

Our minimal customer service requirements are that in any 2 hour window, the GOS should exceed 80%. We will require that this constraint be satisfied with probability 90%, i.e., that the constraint be satisfied on 90% of all days.

The rostering problem is typically solved in two phases [6]. If we assume that the day has been broken into a number of periods, perhaps each of 15 minutes duration, then the first phase is to determine the staffing requirements for each period of the day. We term these staffing levels the *work requirement*. We seek a work requirement that is ‘minimum’ in the sense that the service target cannot be met if staffing is reduced in any period. In the second phase, one attempts to build staff rosters that "cover" the minimum staffing requirements determined in the first phase.

The work requirement (phase 1) is typically determined using queueing models and/or simulation. Queueing models are most applicable when convergence to steady-state is rapid, so that the assumption that the system is in steady-state in any given period is a reasonable one. Unfortunately, rapid convergence to steady state is *not* typical in call centres, especially when the system is periodically heavily loaded, and thus arrival rates are varying from one period to the next. Another approach is to use simulation to determine staffing levels. This approach is taken in, for example, [1]. Yet another approach is to attempt to numerically calculate or approximate the time-varying distribution of GOS. Strongly related ideas are discussed in [3].

There are several problems with the standard approaches. Firstly, there is typically *linkage* between time periods in the sense that the GOS depends on staffing levels in both the current and previous periods. If steady state models are used to calculate the required staffing levels, then this linkage will be ignored, and thus the calculated staffing levels may not yield a feasible solution. Secondly, the linkage between periods typically increases the number of minimum work requirements, i.e. it may be possible to change the staffing requirements in such a way that customer service remains satisfactory, but the solution is still a minimum (in the sense given above). Clearly, where there are many such minimum work requirements, we would like to choose that work requirement that is "easier" to roster, in the sense that the roster generated in Phase 2 is cheaper, or of higher quality. While the final cost/quality of a work requirement is difficult to determine without solving the Phase 2 optimisation problem, some estimate of these measures would be useful in directing the search for a good minimum work requirement.

Our approach attempts to remedy these deficiencies in two ways. First, we attempt to capture some of the linkage effects between time periods in the model used to determine work requirements. To achieve this, we use steady-state queueing models to identify the GOS if the system were run under constant conditions for a sufficiently long period of time. These steady-state results are then modified using heuristics to attempt to capture the time-dependent effects. Secondly, a dynamic program incorporating these time-dependent approximations is solved to obtain minimum staffing levels throughout the day. The cost structure of the dynamic program is designed to construct staffing levels that are "easy" to roster.

Another approach that attempts to capture the linkage between time periods is discussed in [5, 2]. This approach utilizes simulation and integer programming in concert to obtain optimal rosters. The method shows great promise but is in a very early stage of development.

The remainder of this paper is organized as follows. In Section 2, we introduce a queueing model of call centre operation, and provide steady-state results. We also discuss how the GOS requirements are approximated. In Section 3, we explain how the steady-state results of Section 2 are modified to obtain approximations to time-dependent grade of service. In Section 4, we discuss the dynamic program that is used to determine the minimum staffing levels.

2. Steady-State Results

Consider the following model of a call centre. Calls arrive to the call centre according to a non-stationary Poisson process with a piecewise constant arrival rate ($\lambda(t)$: $t \geq 0$). We assume, for definiteness, that the function $\lambda(.)$ is right-continuous. If a server is free, the customer immediately enters service. Service times are exponentially distributed with mean μ^{-1}. If no server is available, the customer queues for service. Customers who are still in the queue when their abandonment time is reached leave the system without ever receiving service. Abandonment times are exponentially distributed with mean α^{-1}. All arrival times, service times and abandonment times are independent.

Under these assumptions, the model is a birth-death process with non-stationary arrival rates. However, if the arrival rate were constant and not time-varying, then one could apply standard results on birth-death processes to obtain stationary distributions. Let $q(.; \lambda, \mu, \alpha, c)$ denote the steady-state distribution of the number of customers in the system when the arrival rate is constant and equal to λ, the service rate is μ, the abandonment rate is α, and there are c servers, so that $q(n; \lambda, \mu, \alpha, c)$ is the steady-state probability that there are n customers in the system. This distribution may be obtained using standard birth-death approaches; see [8] for details.

It is also possible to obtain the steady-state probability p(λ, μ, α, c) that a customer abandons, and the steady-state distribution function w(.; λ, μ, α, c) of the waiting time in the queue conditional on the customer not abandoning. See [8] for details.

These steady-state results may be used to approximate the required GOS statistics as follows. Let $A_i = 1$ if the ith customer abandons, and 0 otherwise. Let W_i be the ith customer's waiting time in the queue. The ith customer's CGOS is then given by CGOS(A_i, W_i), where the function CGOS(.,.) is defined (for example) by

$$CGOS(a,w) = \begin{cases} 0 & if \;\; a = 1 \\ 100 & if \;\; a = 0, w < 10 \\ 80 & if \;\; a = 0, 10 \le w < 30 \\ & \vdots \end{cases}$$

The GOS as measured by VoiceTech is broken down into 2 hour periods of the day. The GOS over a 2 hour period is defined as the average CGOS of customers that arrive in that period. Using a steady-state approximation, the GOS in any period may be approximated by

$$\begin{aligned} GOS &\approx E\ CGOS(A,W) \\ &= 0\ P(A=1) + 100 P(W < 10 \mid A = 0) + \ldots \end{aligned}$$

where the expectation and probabilities are with respect to the stationary distribution with appropriately chosen parameters.

Notice that this approximation to GOS over the 2 hour period may be expressed as a function of the conditional distribution of waiting time given that the customer does not abandon, i.e., the approximate GOS = f(w(.; λ, μ, α)), where f is a known function.

3. Time-Dependent Approximations

As discussed in the previous section, we can approximate the GOS over a given period by

$$\text{GOS} \approx f(w(.; \lambda, \mu, \alpha, c)),$$

where λ, μ, α and c are the parameters for the period in question. However, this approximation assumes the system is in steady-state during the time period in question. We will now extend this approximation using a heuristic approach.

Let $t_0 = 0$, and for $k \ge 1$, define $t_k = \inf\{t > t_{k-1}: \lambda(t) \neq \lambda(t-)\}$, so that the times t_k are the times when the arrival rate changes. For the purposes of exposition, we will assume that the time intervals $(t_k - t_{k-1}: k \ge 1)$ are all of equal length, although this is certainly not necessary for our approach. Let $\tilde{W}_k$ be the waiting time distribution of a randomly selected customer that arrives in the interval $[t_{k-1}, t_k)$. Assuming that the system is empty at time 0, it is reasonable to define $\tilde{W}_k = 0$ (the distribution that is a point mass at 0) for k = 0.

If we could modify t_k, then as t_k got large (with t_{k-1} held fixed), the distribution $\tilde{W}_k$ would converge to $w(.; \lambda(t_{k-1}), \mu, \alpha, c_k)$, where c_k is the number of servers available in period k. This suggests that we might approximate the distribution $\tilde{W}_1$ by a convex combination of W_0 (an approximation to $\tilde{W}_0$) and $w(.;\lambda(0), \mu, \alpha, c_1)$, so that

$$\tilde{W}_1 \approx W_1 = \delta W_0 + (1-\delta) w(\cdot;\lambda(0),\mu,\alpha,c_1),$$

and similarly we could approximate $\tilde{W}_k$ by

$$\tilde{W}_k \approx W_k = \delta W_{k-1} + (1-\delta) w(\cdot;\lambda(t_{k-1}),\mu,\alpha,c_k).$$

Note that this approximation is far from exact. Our goal is a simple modification to the steady-state results given in the previous section. See [4] for details on the theory of convergence to steady-state for Markov processes, of which our model is a special case.

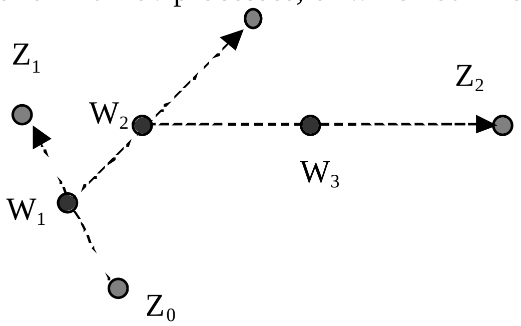

Figure 1: The time-dependent approximation in the space of distributions.

The approximation is depicted in Figure 1. The stationary distributions $W_0 = Z_0$, and ($Z_k = w(.; \lambda(t_{k-1}), \mu, \alpha, c_k)$: $k \geq 1$) are depicted as points in the space of waiting time distributions. The process starts out at time t = 0 at the point Z_0 corresponding to a guaranteed zero waiting time in the queue. If t_1 were to increase, we would expect the distribution $\tilde{W}_1$ to approach Z_1. Our approximation assumes that this convergence occurs along the straight line joining Z_0 and Z_1, and further assumes a particular position along the line (through the parameter δ). Therefore the approximation W_1 lies somewhere on the line segment $[Z_0, Z_1]$. In the following interval $[t_1, t_2)$, the distribution begins to converge to Z_2. Again we take a step along the line joining the current point and Z_2. This process repeats recursively.

We can then approximate the GOS over the period $[t_{k-1}, t_k)$ by $f(W_k)$, where f is the function discussed in Section 2, and W_k is the approximation to the waiting time distribution in the interval $[t_{k-1}, t_k)$. Notice that the function f is linear, in the sense that

$$f(\delta V_1 + (1-\delta) V_2) = \delta f(V_1) + (1-\delta) f(V_2),$$

where $0 \leq \delta \leq 1$, and V_1 and V_2 are 2 waiting time distributions. Hence, the approximate GOS in period k is a convex combination of the previous period's GOS approximation and the GOS corresponding to the stationary distribution $w(.; \lambda(t_{k-1}), \mu, \alpha, c_k)$.

The final step in the approximation is to choose the step parameter δ. For small time intervals, δ should be chosen close to 1 so that a small step is taken towards the new stationary distribution, whereas for large time intervals δ should be chosen close to 0. See [8] for further details on an appropriate choice of δ.

The approximation to the GOS in each period is very easily calculated for a given set of server allocations (c_k: $k \geq 1$). In particular, the GOS in the next period k+1 can be easily calculated given the current period k’s distribution and the number of staff c_{k+1} on duty in the next period. More importantly, because f is linear, the actual distibution is not required, but only its GOS. Thus, the next period's GOS can be calculated from the current period's GOS and the next period's staffing level c_{k+1}. This observation, depicted in Figure 2, underpins the dynamic program that attempts to select suitable c_k's.

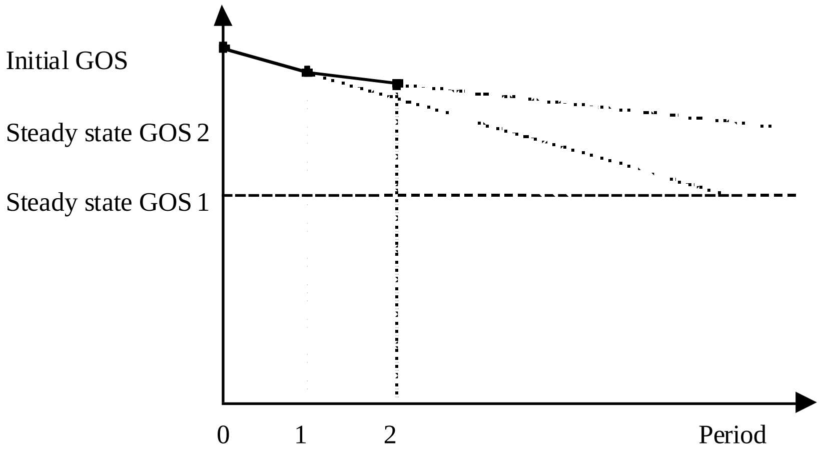

Figure 2: Schematic of the sequence of distributions viewed in the GOS space.

Staffing Dynamic Program

Our objective with the staffing dynamic program (DP) is to determine staffing levels for the periods that together provide a minimum work requirement that meets the GOS target for each period and will also result in a good staff roster. A schematic of this DP is shown below in Figure 3. The periods in the day (being typically of 15 minutes duration) form the stages of the DP; these are plotted on the horizontal axis. Each state in the DP – shown on the vertical axis - defines a possible grade of service that may be achieved in the associated period (stage). An arc [k,f,c] in the DP represents a change in the GOS from one period k to the next period, period k+1, for a given starting GOS f (the state shown at the tail of the arc) and a specified staffing level c (shown beside the arc). The resultant GOS for the arc, $f_k(f,c)$, is calculated from the approximation discussed above using the arrival rate associated with period k.

From this example it can be seen that the system starts with 100% GOS in stage 0. By making differing numbers of agents available for service we move to the corresponding GOS for that staffing level in stage 1 (the end of the first quarter-hour period). It should be noted that not every GOS state can be reached as there are only an integer number of agents. A similar process is repeated from stage 1, where we only examine the path forward from states that have been reached from stage 0. It can be seen that the state associated with a GOS value of 80% in stage 2 can be reached by two distinct paths. Thus a decision needs to be made about which is the better path to take. This decision is made by comparing the cumulative *costs* of the arcs on each path, where the cost on arc [k,f,c] is an estimate of the rostering consequences of requesting c staff in period k. These costs attempt to penalise peaks in the work requirement as these are typically difficult to cover in the rostering phase. The structure of these costs is discussed further in [7] and [8].

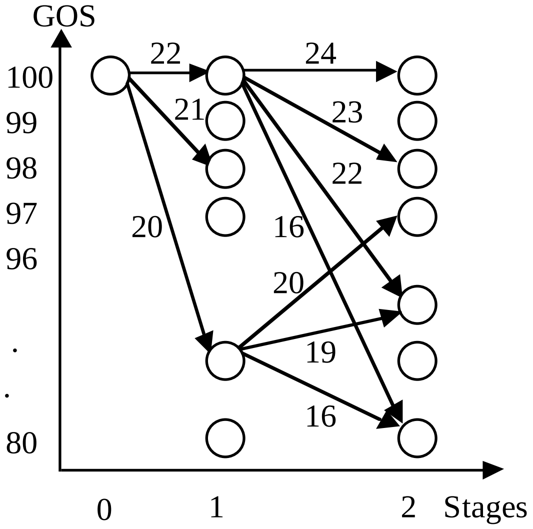

Figure 3: Schematic of the DP stage/state space.

To ensure that the minimum permitted GOS of 80% is achieved in each period, any arcs that result in a GOS less than 80% are banned. Thus, any solution that produces a GOS value less than 80% is deemed infeasible and is not considered in the solution process. For computational efficiency, the GOS state space was discretised into unit segments and any resultant GOS $f_k(f,c)$ was truncated to fit these partitions. This meant we in fact had only 21 possible states (80 -100%) for the GOS.

There are a number of simple extensions that can be made to this model. Firstly, the DP state space can be expanded to include a cumulative GOS sum taken over all calls in the preceeding periods. This allows a daily average GOS to be specified in addition to the GOS target for each period. For example, the user may require an average GOS of at least 85% over the day’s calls with the GOS in any period being at least 80%. A second extension is to ban staffing solutions that exceed some specified maximum abandonment rate in any period. These abandonment rates can be calcuated from the time-dependent distribution approximation, and effectively ban possible GOS levels in each stage.

GOS Distribution

In the above discussion, we have only considered expected values for the GOS. If the GOS distribution is symmetric then we could state that on 50% of days the GOS will be under our target. From testing with a simulation, we have observed that the GOS can be modelled by a normal distribution as shown in Figure 4. However as the mean GOS value tends towards 100%, the distribution becomes skewed due to the limit that the GOS can be at most 100%. The GOS distribution is then better modeled using a beta distribution. However in the area of interest, around the 10th - 50th percentiles, both the normal approximation and the beta approximation produce similar results. Using the normal approximation, we can now increase the GOS target seen by the DP to ensure we meet our real target with a user-specified probability that is typically 90%.

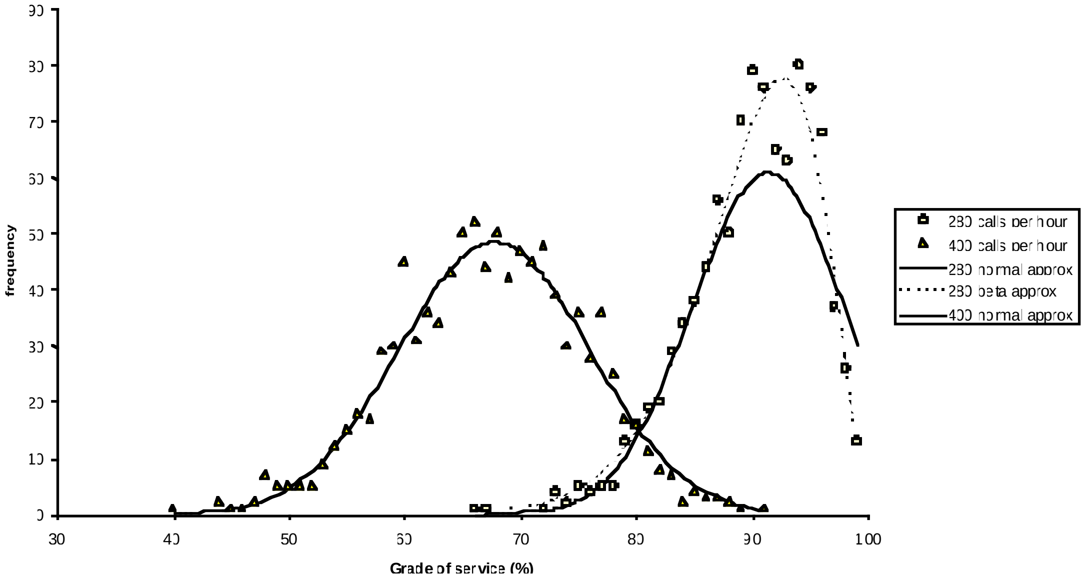

Figure 7-5: Normal approximation to the GOS distribution.

Results

Although commercial sensitivity prohibits publication of detailed results, we can reveal that simulation results confirm that the work requirements produced by the DP model are indeed feasible and behave as predicted by the normal approximation. Further work is continuing to implement rostering systems that can turn the work requirements into actual worked rosters, and thus demonstrate the full value of this new approach.

Acknowledgments

This research was supported through a Graduates in Research Fellowship from the Public Good Science Fund. We would like to thank VoiceTech Systems Ltd for their support of this project.

References

[1] Eitzen, G.E. 1994. Telephone resource allocation problem. Honours Thesis, School of Mathematics, University of South Australia.

[2] Henderson, S., and A. Mason. 1998. Rostering by iterating integer programming and simulation. *Proceedings of the 1998 Winter Simulation Conference. D. Medeiros, E. Watson, eds.* IEEE.

[3] Jennings, O., A. Mandelbaum, W. Massey, and W. Whitt. 1996. Server staffing to meet time varying demand. *Management Science* 42: 1383 - 1394.

[4] Lindvall, T. 1992. *Lectures on the Coupling Method*. Wiley, New York.

[5] Mason, A., and S. Henderson. 1998. Call Centre Rostering by Iterating Integer Programming and Simulation. *Proceedings of the 33rd Conference of the Operational Research Society of New Zealand*.

[6] Mehrotra, V. 1997. Ringing up big business. *OR/MS Today* 24(4).

[7] Rawles, R. (1996) A Dynamic Programming Engine for Determining Customs Staffing Requirements at Auckland Airport. A Fourth Year Project, Department of Engineering Science, School of Engineering, University of Auckland, pp50.

[8] Thomson, R. 1998. *Decision Support for Call Centre Design and Management.* Master's Thesis, Department of Engineering Science, University of Auckland, New Zealand.