A note on profit maximization and monotonicity for inbound call centers

Ger Koole & Auke Pot
Department of Mathematics, Vrije Universiteit Amsterdam, The Netherlands

23rd December 2005

Abstract

We consider an inbound call center with a fixed reward per call and communication and agent costs. By controlling the number of lines and the number of agents we can maximize the profit. Abandonments are included in our performance model. Monotonicity results for the maximization problem are obtained, which leads to an efficient optimization procedure. We give a counterexample to the concavity in the number of agents, which is equivalent to saying that the law of diminishing returns does not hold. Numerical results are given.

1 Introduction

Traditionally call centers are seen as cost centers. This means that a certain service level has to be obtained for minimal costs. The service level is often taken as follows: 80% of the calls should be answered by a call center agent within 20 seconds. In this business model a call costs money but is necessary to the company.

Another business model is the one where the call center is considered to be a profit center. There is a reward assigned to a call, and the objective of the call center is to maximize its profit, defined as rewards minus costs. The variable costs in a call center mainly consist of salary costs and communication costs, in case the call center pays for (part of) that. Such a business model can lead to considerable savings (see [2] for an example).

In this paper we analyze a model for a call center in which we determine the number of lines and agents for which the profit is maximized. It is our objective to find the global maximum for this two-dimensional profit function. We derive properties of the profit function as to avoid having to search exhaustively all possible combinations of parameter values. Based on this we formulate an algorithm that finds the global maximum. The optimization procedure can be seen as a local search procedure in which the function value is the profit for given parameter values. These values can be obtained through Markov chain methods. We give some numerical results. The interested reader can also experiment with a tool that is freely available on the internet.

In addition to the positive results that lead to the optimization procedure we also offer a counterexample that shows that (given the optimal number of lines is determined for each possible

number of agents) the profit function is neither concave nor unimodal in the number of agents. This shows that the 'law of diminishing returns' does not hold for this model.

Our model is one of the models studied in Helber et al. [4] (see also [3]). In this paper the authors survey the German call center market and come up with a number of business models. Next to our single period profit maximization model they also consider a multi-period model with a constraint on the number of available agent hours. Our monotonicity results show how to find the optimal solution in the single-period model of [4], and our counterexample shows that finding the optimal solution for the multi-period model is, in theory, a non-trivial problem. A different call center profit maximization model (with multiple call classes and a shared resource) has been described in Akşin and Harker [1].

In the next section we describe the model. After that we present the properties of the profit function and the resulting optimization procedure. We also give numerical results. The section after that is devoted to the counterexample and its implications. In the final section the properties of the profit function are derived, mainly using dynamic programming. This is done by inductively proving certain properties, related to concavity, of the dynamic programming value function.

2 Model description and results

The call center model that we consider is commonly called an M/M/s/n+M system. That is, it has Poisson arrivals and exponential service times, with additional features exponential abandonments and a finite number of lines meaning that the total number of calls waiting and in service is restricted. There are no redials of blocked or abandoned calls.

The customer arrival rate is λ, and the rate of the service time distribution is μ. (In practice usually the expected call duration is taken, which we denote with $\beta = 1/\mu$.) The number of agents is variable, with an upper bound of S, the number of 'seats' in the call center. Also the number of lines is limited, to $s+N$ if there are s agents. (We take $s+N$ instead of simply N for reasons that will become clear later.) A call that is waiting abandons with rate γ.

We have communication costs c per call per unit of time, and costs 1 per scheduled agent per unit of time. There are expected rewards r per handled call. (Note that the actual reward per call might vary, but as we are only interested in expected rewards, and as we have no prior information on the reward of a call, we only need r.)

We define $g^{s,n}$ as the average long-run expected profit for s agents or servers and n additional waiting lines. For fixed s and n $g^{s,n}$ is the stationary reward in a birth-death process with states $x \in \{0,\ldots,s+n\}$ (indicating the number of calls in the system), transition rates $\alpha(\cdot,\cdot)$, and immediate rewards $\delta(\cdot)$, given by:
$\alpha(x,x+1) = \lambda$ for $0 \le x < s+n$, $\alpha(x,x-1) = \min\{x,s\}\mu + (x - \min\{x,s\})\gamma$ for $0 < x \le s+n$;
$\delta(x) = \min\{x,s\}\mu r - xc - s$.

Using standard arguments for birth-death processes it follows that $g^{s,n}$ is given by (take $a = \lambda\beta$):

$$g^{s,n} = \frac{\sum_{x=0}^{s} \frac{a^x}{x!} x(\mu r - c) + \frac{a^s}{s!} \sum_{x=1}^{n} \frac{\lambda^x}{\prod_{y=1}^{x}(s\mu + y\gamma)} (s(\mu r - c) - xc)}{\sum_{x=0}^{s} \frac{a^x}{x!} + \frac{a^s}{s!} \sum_{x=1}^{n} \frac{\lambda^x}{\prod_{y=1}^{x}(s\mu + y\gamma)}} - s. \tag{1}$$

Now define $g^s = \max_{0 \le n \le N} g^{s,n}$ (with $n_s = \arg\max_{0 \le n \le N} g^{s,n}$) and $g = \max_{0 \le s \le S} g^s$ (with $s^* = \arg\max_{0 \le s \le S} g^s$). In Section 4 we show the following results.

Theorem 2.1 *$g^{s,0} \le \cdots \le g^{s,n_s}$ for all $0 \le s \le S$ and $n_s \le n_{s+1}$ for all $0 \le s < S$.*

Theorem 2.1 tells us the following: for a fixed number of agents the reward is non-decreasing in the number of lines up to the optimal number of lines, and if the number of agents is increased than the optimal number of lines for that many servers does not decrease.

The theorem leads to a simple algorithm for finding s^* and n_{s^*}. To avoid trivialities we check first that $(1+c)\beta < r$: if this is not the case then the costs for the agent and communication of a call that is directly connected are higher than the profit, and it is better to reject all calls and to schedule no agents at all.

Algorithm for finding (s^*, n_{s^*}):
0. Take $(s,n) = (s^*, n_{s^*}) = (0,0)$
1. If $(1+c)\beta \ge r$ then: stop
2. For $s = 1$ to S do
3. Compute $g^{s,n}$ (using Equation (1))
3. If $n < N$ then: Compute $g^{s,n+1}$
4. While $g^{s,n} < g^{s,\min\{n+1,N\}}$
5. $n \leftarrow n+1$
6. If $n < N$ then: Compute $g^{s,n+1}$
7. If $g^{s,n} > g^{s^*,n_{s^*}}$ then: $(s^*, n_{s^*}) \leftarrow (s,n)$

Thus we see that for each value of s we increase n until we have found the optimal value n_s; then we increase s, and we start increasing n again, from the value n_s. According to Theorem 2.1 we certainly encounter the optimal solution, but we can only identify it after having determined n_s for all values of s, because g^s need not be unimodal, as the counterexample in the next section shows.

In Figure 1 we see a typical example of how the algorithm traverses the (s,n)-grid, from the lower left corner to the upper right. The corresponding parameters are $S = 10$, $N = 30$, $\lambda = 5$, $\mu = 1$, $c = .5$, $r = 3$, $\gamma = .5$, and the price of an agent 1 per time unit. It takes at maximum $S+N$ steps, while there are SN points in the grid. This illustrates well the efficiency of the algorithm as compared to enumeration. It makes it suitable for routine application to typical call center

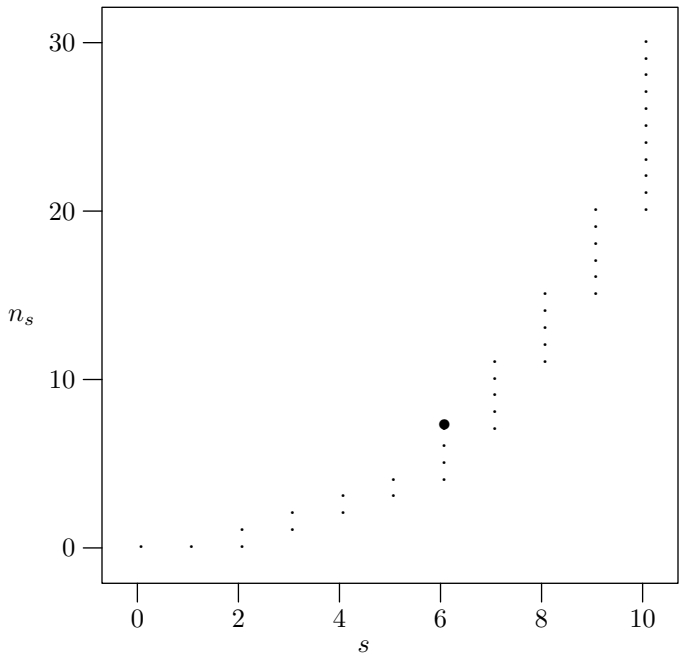

Figure 1: (s,n)-grid for $S = 10$, $N = 30$, $\lambda = 5$, $\mu = 1$, $c = .5$, $r = 3$, and $\gamma = .5$

optimization problems with tens of different intervals with different parameters per day. The optimum is $(s^*, n_{s^*}) = (6, 13)$.

The tool by which the numerical results were obtained can also be found on our web site, see www.math.vu.nl/∼sapot/software/ErlangProfit/. Note that in the tool the number of lines includes the number of agents, which is more usual in practice. Also the average service and abandonment times have to be entered, not the rates.

3 Counterexample and implications

In case g^s were unimodal then we could stop searching as soon as g^s would decrease after some s. We show by a counterexample that this is not always the case. Consider the model with the parameters $\lambda = 15$, $\mu = 1$, $c = 0.39$, $r = 1.52$, $\gamma = 1/2.9$ and an agent costs 1 per unit of time, as we defined earlier.

To analyze the concavity we vary s from 0 to 15. In Table 1 the values of g^s can be found. We see that g^s increases for s up to $s = 8$, then it decreases for $s = 9$, to take its maximum values at $s = 10$. We conclude that the function g^{s,n_s} is non-unimodal and thus neither convex nor concave in s. This counterexample shows the necessity of increasing s up to S in the algorithm.

The intuition behind it is as follows. Computations show that $n_8 = 1$ and $n_9 = n_{10} = 2$. Thus when adding the 9th server it is optimal to add an additional line for waiting. However, for $n = 2$, it is better to have 10 instead of 9 servers. Thus $s = 9$ does not justify completely the second line, but with a single line the productivity and thus the reward is too low ($g^{9,1} = 0.3824$). Thus due

s	g^s
0	0.0000
1	0.0594
2	0.1105
3	0.1521
4	0.1825
5	0.2396
6	0.3147
7	0.3665
8	0.3907
9	0.3855
10	0.3993
11	0.3636
12	0.2951
13	0.1771
14	0.0033
15	−0.2561

Table 1: Values of g^s for various s for $\lambda = 15$, $\mu = 1$, $c = 0.39$, $r = 1.52$, and $\gamma = 1/2.9$

to the discrete nature of n we see a drop in profit at $s = 9$.

The counterexample has further implications. Consider we have two (or more) intervals and a limited number of agent hours, as in [4]. How to allocate the agent hours in an optimal way? A greedy algorithm would find the optimal allocation, assuming that the 'law of diminishing returns' holds. This law states that when the number of servers is increased the additional returns for adding servers decrease. This is equivalent to concavity. Indeed, for a sum of concave functions the optimal allocation can be found by starting empty and adding agent hours one by one, each time to the interval with the highest additional return. It follows clearly from the counterexample that the concavity does not hold, and thus the greedy algorithm is not guaranteed to give an optimal solution in the multi-period model of [4].

4 Monotonicity results

In this section we prove Theorem 2.1.

Proof of Theorem 2.1 We start with proving $g^{s,0} \leq \cdots \leq g^{s,n_s}$ for some s with $0 \leq s \leq S$. Assume that $n_s > 0$ (otherwise there is nothing to prove). First note that $g^{s,n+1} = pg^{s,n} + (1-p)\delta(s+n+1)$ for some $0 < p < 1$. This equation holds for all birth-death processes. Now suppose that $g^{s,n} > g^{s,n+1}$ for some $n < n_s$. This means that $\delta(s+n+1) < g^{s,n}$. Note also that $\delta(s+n_s) < \cdots < \delta(s+n+1)$. Because g^{s,n_s} is a convex combination of $\delta(s+n_s), \ldots, \delta(s+n+1)$ and $g^{s,n}$, this means that $g^{s,n_s} < g^{s,n}$, which is in contradiction with the optimality of n_s.

The proof of the second assertion of Theorem 2.1 is more involved. We use dynamic programming in its proof. We formulate the value function for fixed s and admission control. It is well known that a threshold policy is optimal (Lippman [6]; see Koole [5] for an overview of this type of monotonicity result). We called this threshold n_s, meaning that an arrival is rejected if and only if the number of customers exceeds $s+n_s$. To prove $n_s \leq n_{s+1}$ for some $0 \leq s < S$ we need to show that when admission is optimal in the system with s servers and a total of x customers, then admission is also optimal in the system with $s+1$ servers and a total of $x+1$ customers (giving the same number of waiting customers). Let us now formulate the dynamic programming value function. We scale time such that $\lambda + S\mu + N\gamma = 1$. Then the transition rates can also be seen as transition probabilities of the embedded uniformized chain (see Lippman [6]). The dynamic programming value function V_k^s of this embedded chain, with k the epoch, is now given by

$$V_{k+1}^s(x) = \mu r \min\{s,x\} - cx - s + \lambda \max\{V_k^s(x), V_k^s(x+1)\} +$$

$$[\mu \min\{s,x\} + \gamma(x - \min\{s,x\})]V_k^s(x-1) + [\mu(S - \min\{s,x\}) + \gamma(N - x + \min\{s,x\})]V_k^s(x)$$

if $x < s+N$ and $k > 0$,

$$V_{k+1}^s(s+N) = \mu r s - c(s+N) - s + \lambda V_k^s(s+N) + [\mu s + \gamma N]V_k^s(s+N-1) +$$

$$\mu(S-s)V_k^s(s+N)$$

if $k > 0$, and $V_0^s(x) = 0$ for all s and x. Note that final term in both equations comes from the uniformization procedure.

To prove that admission is optimal in the system with $s+1$ servers and $x+1$ customers if it is optimal in the system with s servers and x customers, it suffices to show that

$$V_k^s(x+1) + V_k^{s+1}(x+1) \leq V_k^s(x) + V_k^{s+1}(x+2) \tag{2}$$

for all $k \geq 0$, $0 \leq s < S$, and $0 \leq x < s+N-1$. Indeed, if admission is optimal in x when having s servers, and thus $V_k^s(x) \leq V_k^s(x+1)$, then according to Equation (2) also $V_k^{s+1}(x) \leq V_k^{s+1}(x+1)$, and admission is also optimal with x calls and $s+1$ servers or agents. From Markov decision theory it follows that the same holds for the long-run limiting average case (because state and action spaces are finite: see, e.g., Puterman [7]).

In the proof of inequality (2) we will use the well-known concavity of V_k^s in x, i.e.,

$$V_k^s(x) + V_k^s(x+2) \leq V_k^s(x+1) + V_k^s(x+1). \tag{3}$$

Note that inequalities (2) and (3) summed gives supermodularity (or convexity).

$$V_k^s(x+1) + V_k^{s+1}(x) \leq V_k^s(x) + V_k^{s+1}(x+1) \tag{4}$$

We are now ready to prove (2). We do this by induction to k. For $k = 0$ the inequality trivially holds. Assume that it holds up to some k. Now consider the corresponding terms in V_{k+1}^s and

V_{k+1}^{s+1} one by one (a method formalized in [5]). Consider first the rewards, the inequality to show is

$$\min\{s,x+1\}+\min\{s+1,x+1\} \leq \min\{s,x\}+\min\{s+1,x+2\}.$$

It is readily shown that this holds indeed for all values of x and s. The same holds for the costs. Now consider the term with coefficient λ. We have to look at a number of cases. Assume first that the maximizing action in both $V_k^s(x+1)$ and $V_k^{s+1}(x+1)$ is admission. Then

$$\max\{V_k^s(x+1),V_k^s(x+2)\}+\max\{V_k^{s+1}(x+1),V_k^{s+1}(x+2)\} = V_k^s(x+2)+V_k^{s+1}(x+2) \leq$$

$$V_k^s(x+1)+V_k^{s+1}(x+3) \leq \max\{V_k^s(x),V_k^s(x+1)\}+\max\{V_k^{s+1}(x+2),V_k^{s+1}(x+3)\},$$

the first inequality is obtained by induction. A similar argument holds in the case that rejection is optimal in state $x+1$ for the systems with s and $s+1$ servers. If the optimal actions are different, then it must be that admission is the optimizing action in $V_k^{s+1}(x+1)$, by induction. Then

$$\max\{V_k^s(x+1),V_k^s(x+2)\}+\max\{V_k^{s+1}(x+1),V_k^{s+1}(x+2)\} = V_k^s(x+1)+V_k^{s+1}(x+2) \leq$$

$$\max\{V_k^s(x),V_k^s(x+1)\}+\max\{V_k^{s+1}(x+2),V_k^{s+1}(x+3)\}.$$

Consider next the terms with coefficient μ, the departure terms. There coefficients sum up to $S\mu$, as if there are in total S servers. For value function V_k^s s of these are present, and in state x $\min\{s,x\}$ of these are active. We number the servers, and assume that in state x servers 1 upto $\min\{s,x\}$ are active. We consider the servers one by one. Assume first that $s \geq x+1$. Then all calls in (2) are being served. Of particular interest are server $x+1$ and $x+2$, the terms related to all other servers hold trivially by induction. Server $x+1$ leads on the l.h.s. to $V_k^s(x)+V_k^{s+1}(x)$, on the r.h.s. to $V_k^s(x)+V_k^{s+1}(x+1)$. Server $x+2$ leads on the l.h.s. to $V_k^s(x+1)+V_k^{s+1}(x+1)$, on the r.h.s. to $V_k^s(x)+V_k^{s+1}(x+1)$. Both l.h.s. summed are smaller than the r.h.s. summed, because of Equation (4), which holds by induction. Next assume that $s \leq x$. Then all servers are busy, and the $s+1$th server gives again Equation (4).

Finally consider the abandonments. Assume that $s \leq x$, otherwise there are no abandonments. Taking abandonment of the extra customer in queue in $V_k^s(x+1)$ and $V_k^{s+1}(x+2)$ into account leads to an equality. □

References

[1] O.Z. Akşin and P.T. Harker. Capacity sizing in the presence of a common shared resource: Staffing an inbound call center. *European Journal of Operational Research*, 147:464–483, 2003.

[2] B. Andrews and H. Parsons. Establishing telephone-agent staffing levels through economic optimization. *Interfaces*, 23(2):14–20, 1993.

[3] S. Helber and R. Stolletz. *Call Center Management in der Praxis*. Springer, 2003.

[4] S. Helber, R. Stolletz, and S. Bothe. Erfolgszielorientierte Agentenallokation in inbound call Centern. *Zeitschrift für Betriebswirtschaftliche Forschung*, pages 3–32, 2005 (February).

[5] G.M. Koole. Structural results for the control of queueing systems using event-based dynamic programming. *Queueing Systems*, 30:323–339, 1998.

[6] S.A. Lippman. Applying a new device in the optimization of exponential queueing systems. *Operations Research*, 23:687–710, 1975.

[7] M.L. Puterman. *Markov Decision Processes*. Wiley, 1994.