OPERATIONS RESEARCH
Vol. 59, No. 5, September–October 2011, pp. 1304–1308
ISSN 0030-364X | EISSN 1526-5463 | 11 | 5905 | 1304

informs

http://dx.doi.org/10.1287/opre.1110.0990

TECHNICAL NOTE

A Note on Profit Maximization and Monotonicity for Inbound Call Centers

Ger Koole
Department of Mathematics, VU University Amsterdam, 1081 HV Amsterdam, The Netherlands, koole@few.vu.nl

Auke Pot
CCmath, 1181 BH Amstelveen, The Netherlands, pot@ccmath.com

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 lead 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.

Subject classifications: call centers; monotonicity.
Area of review: Manufacturing, Service, and Supply Chain Operations.
History: Received July 2008; revisions received January 2009, August 2009; accepted October 2009.

1. Introduction

Traditionally, call centers are seen as *cost centers*. Cost centers add to the costs but not directly to the profit of the company concerned. Therefore, reducing costs is a prime goal in cost centers, given that its products meet the objectives of the company. In the setting of call centers, this means that a certain service level has to be obtained for minimal costs. This 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.

Lately there has been some criticism of considering call centers as cost centers. The focus on, for example, the reduction of handling times leads to low-quality contacts with (potential) customers. To value the customer contacts, call centers should not be treated as cost centers but as *profit centers*. A profit center adds directly to the profit of a company. In a call-center setting, this means that each contact adds value. 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 profit increases.

A perfect example where replacing cost minimization by profit maximization leads to considerable cost savings is described in Andrews and Parsons (1989), Andrews and Parsons (1993). In Andrews and Parsons (1989) the authors compare for a particular case the profit of different agent scheduling tools, who themselves use a service-level constraint in the Erlang C model to obtain staffing levels. In Andrews and Parsons (1993) the agent staffing at interval level is considered. The optimal staffing level is determined by profit maximization: every handled call gives a reward, waiting calls lead to (communication) costs, and every scheduled agent costs a fixed amount. Increasing the number of agents increases agent costs, but decreases waiting costs and decreases the abandonment rate, thereby increasing the reward. It is remarkable that the Erlang C model is used, because abandonments (which are not modeled by Erlang C) play a crucial role in the paper. This is justified by the low abandonment rate in the case study; it is assumed that in that case the influence on the service level is limited (note that this has been contradicted by recent work on the Erlang A model; see, for example, Gans et al. 2003, §4.2.2). The abandonment percentage is obtained by linear regression with the service-level as explanatory variable.

In this paper we study a situation where, in addition to the number of agents, the number of lines can also be determined. In practice this means that arriving calls are blocked and asked to call again at another moment. This avoids excessively long waiting times for calls for which there is no capacity anyhow. It is our objective to find the global maximum for this two-dimensional profit function, using a model that includes abandonments. We derive properties of the profit function so 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 that 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. (2005) (see also Helber and Stolletz 2003). In this paper the authors survey the German call-center market and come up with a number of business models. Besides, to our single-period profit-maximization model they also consider a multiperiod 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 Helber et al. (2005), and our counterexample shows that finding the optimal solution for the multiperiod model is, in theory, a nontrivial problem. A different call-center profit-maximization model (with multiple call classes and a shared resource) has been described in Akşin and Harker (2003).

There is a single condition that is required in part of our proofs: the average service time should be shorter than the average patience (or "willingness to wait") of a call. The reason is a technical one related to convexity of dynamic programming value functions. Independently, Armony et al. (2009) obtained similar results.

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. We use the ideas of event-based programming in this part of the proof (see Koole 2006). In this section we also go into more detail about the restriction on the patience time.

2. Model Description and Results

The call-center model that we consider is commonly called an $M/M/s/n+M$ system. That is, as in the Erlang C model it has Poisson arrivals, exponential service times, and a finite number of servers (s). However, it has two additional features: customers who find all servers occupied leave the queue after an exponentially distributed amount of time (their *patience*), and there is 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 we are only interested in the expected reward r because we have no prior information on the reward of a call.)

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:

$$\alpha(x, x+1) = \lambda \quad \text{for } 0 \leqslant x < s+n,$$
$$\alpha(x, x-1) = \min\{x, s\}\mu + (x - \min\{x, s\})\gamma \qquad \text{for } 0 < x \leqslant s+n;$$
$$\delta(x) = \min\{x, s\}\mu r - xc - s.$$

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

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

Now define

$$g^s = \max_{0 \leqslant n \leqslant N} g^{s,n} \quad (\text{with } n_s = \arg\max_{0 \leqslant n \leqslant N} g^{s,n})$$

and

$$g = \max_{0 \leqslant s \leqslant S} g^s \quad (\text{with } s^* = \arg\max_{0 \leqslant s \leqslant S} g^s).$$

In §4 we show the following results.

Theorem 2.1. *$g^{s,0} \leqslant \cdots \leqslant g^{s,n_s}$ for all $0 \leqslant s \leqslant S$ and, if $\mu \geqslant \gamma$, $n_s \leqslant n_{s+1}$ for all $0 \leqslant s < S$.*

Theorem 2.1 tells us the following: for a fixed number of agents the reward is nondecreasing in the number of lines up to the optimal number of lines, and if the number of agents is increased, then the optimal number of lines for that many servers does not decrease. The latter statement holds only in the case that the service rate is not lower than the abandonment rate.

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.

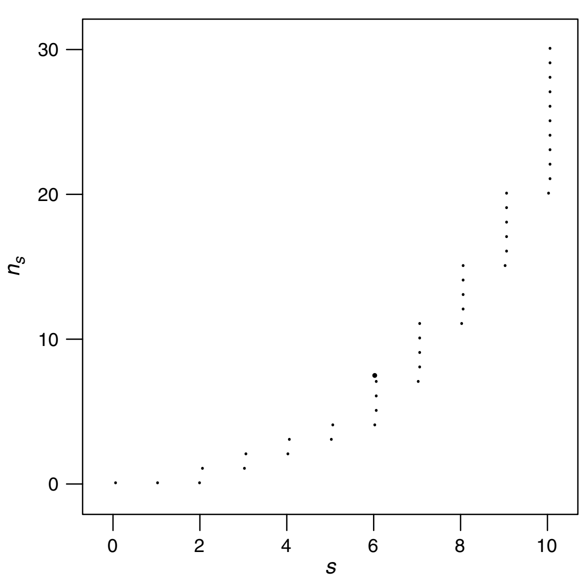
Algorithm for finding (s^*, n_{s^*}):
0. Take $(s, n) = (s^*, n_{s^*}) = (0, 0)$
1. If $(1 + c)\beta \geqslant r$, then: stop
2. For $s = 1$ to S do
3. Compute $g^{s,n}$ (using Equation (1))
3. If $n < N$, then: Compute $g^{s,n+1}$
4. While $g^{s,n} < g^{s,\min\{n+1,N\}}$
5. $n \leftarrow n + 1$
6. If $n < N$, 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 $S = 10$, $N = 30$, $\lambda = 5$, $\mu = 1$, $c = 0.5$, $r = 3$, $\gamma = 0.5$, and the price of an agent 1 per time unit. It takes at maximum $S + N$ steps, whereas 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 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 website; see http://obp.math.vu.nl/callcenters/ErlangP. 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.

Figure 1. (s, n)-grid for $S = 10$, $N = 30$, $\lambda = 5$, $\mu = 1$, $c = 0.5$, $r = 3$, and $\gamma = 0.5$.



An interesting special case is $c = 0$. Then there is no reason to refuse any customers, and thereby the number of lines should be as big as possible. Thus, all that needs to be determined is the optimal number of servers. Now we can apply Theorem 11.7 from Koole (2006), which shows that the reward is concave in s, again under the assumption that $\mu \geqslant \gamma$. Thus, it suffices to increase s until the reward starts decreasing. When $c > 0$, the optimal number of lines is always finite: if the queue is long enough, then the holding costs c outweigh the service reward r.

A final remark concerns the situation where the holding costs are different in service than when waiting. This, however, can be translated to the situation where holding costs are equal by changing r appropriately.

3. Counterexample and Implications

If 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 nonunimodal 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 completely justify the second line, but with a single line the productivity and thus the reward is too low ($g^{9,1} = 0.3824$). Thus, due to the discrete nature of n we see a drop in profit at $s = 9$.

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

s	g^s	n_s	Abandonments (in %)	Blocked (in %)	SL (in %)
0	0.0000	0	0.00	100.00	100.00
1	0.0594	0	0.00	93.75	100.00
2	0.1105	0	0.00	87.55	100.00
3	0.1521	0	0.00	81.40	100.00
4	0.1825	0	0.00	75.32	100.00
5	0.2396	1	4.47	66.05	86.91
6	0.3147	1	3.45	59.99	91.42
7	0.3665	1	2.70	54.05	94.41
8	0.3907	1	2.15	48.28	96.39
9	0.3855	2	4.02	39.79	90.53
10	0.3993	2	3.26	34.36	93.56
11	0.3636	2	2.64	29.22	95.71
12	0.2951	3	3.54	21.94	92.46
13	0.1771	3	2.81	17.62	94.97
14	0.0033	4	3.13	11.86	93.21
15	−0.2561	5	3.09	7.29	92.49

The counterexample has further implications. Consider we have two (or more) intervals and a limited number of agent hours, as in Helber et al. (2005). 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 multiperiod model of Helber et al. (2005).

It is interesting to note that this is not the only situation where admission control destroys the concavity. Take the regular M/M/s queue: it can be shown that the value function is concave in the arrival rate for many common situations. However, a counterexample to concavity exists as soon as admission control is added. For more details, see Varga (2002) and the last part of §11.2 of Koole (2006).

4. Monotonicity Results

In this section we first prove Theorem 2.1. Then we discuss some issues related to convexity.

Proof of Theorem 2.1. We start with proving $g^{s,0} \leqslant \cdots \leqslant g^{s,n_s}$ for some s with $0 \leqslant s \leqslant S$. 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 1975; see Koole 2006 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 $n_s \leqslant n_{s+1}$ for some $0 \leqslant s < S$, 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 1975). Costs and rewards are implemented as follows. When a customer is allowed to enter, we receive a reward of r. When a customer abandons, we incur a cost of r. In state x this happens at rate $\gamma(x-s)^+$. Next to that there are communication costs and server costs $cx+s$. The dynamic programming value function V_k^s of the embedded chain, with k the epoch, is now given by

$$\begin{aligned}
&V_{k+1}^s(x)\\
&\quad = -\gamma r(x-s)^+ - cx - s + \lambda \max\{V_k^s(x), r + V_k^s(x+1)\}\\
&\qquad + [\mu \min\{s,x\} + \gamma(x-s)^+]V_k^s(x-1)\\
&\qquad + [\mu(S - \min\{s,x\}) + \gamma(N-(x-s)^+)]V_k^s(x)\\
&\qquad\qquad \text{if } x < s+N \text{ and } k > 0,\\
&V_{k+1}^s(s+N)\\
&\quad = -\gamma rN - c(s+N) - s + \lambda V_k^s(s+N)\\
&\qquad + [\mu s + \gamma N]V_k^s(s+N-1) + \mu(S-s)V_k^s(s+N)\\
&\qquad\qquad \text{if } k > 0, \text{ and } V_0^s(x) = 0 \text{ for all } s \text{ and } x.
\end{aligned}$$

Note that the 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

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

for all $k \geqslant 0$, $0 \leqslant s < S$, and $0 \leqslant x < s+N-1$. Indeed, if admission is optimal in x when having s servers, and thus $V_k^s(x) \leqslant r + V_k^s(x+1)$, then according to Equation (2), also $V_k^{s+1}(x+1) \leqslant r + V_k^{s+1}(x+2)$, and admission is also optimal with $x+1$ 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 1994).

In the proof of inequality (2) we need concavity of V_k^s in x, i.e.,

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

This result is valid only if $\mu \geqslant \gamma$; see Koole (2006, Theorem 9.3) and the discussion in the last paragraph on p. 50. (Note that in Koole 2006 a cost setting is used, interchanging convexity and concavity for the value function.)

Summing inequalities (2) and (3) with s replaced by $s+1$ leads to supermodularity:

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

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 first formalized in Koole 1998, see also Koole 2006). We start with the first term of the dynamic programming equation, the one related to the rewards. The inequality to show is

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

It is readily shown that this indeed holds 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

$$\begin{aligned}&\max\{V_k^s(x+1), V_k^s(x+2)\} + \max\{V_k^{s+1}(x+1), V_k^{s+1}(x+2)\}\\ &= V_k^s(x+2) + V_k^{s+1}(x+2) \leqslant V_k^s(x+1) + V_k^{s+1}(x+3)\\ &\leqslant \max\{V_k^s(x), V_k^s(x+1)\} + \max\{V_k^{s+1}(x+2), V_k^{s+1}(x+3)\},\end{aligned}$$

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

$$\begin{aligned}&\max\{V_k^s(x+1), V_k^s(x+2)\} + \max\{V_k^{s+1}(x+1), V_k^{s+1}(x+2)\}\\ &= V_k^s(x+1) + V_k^{s+1}(x+2)\\ &\leqslant \max\{V_k^s(x), V_k^s(x+1)\} + \max\{V_k^{s+1}(x+2), V_k^{s+1}(x+3)\}.\end{aligned}$$

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 up to $\min\{s, x\}$ are active. We consider the servers one by one. Assume first that $s \geqslant x+1$. 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 left-hand side to $V_k^s(x) + V_k^{s+1}(x)$, on the right-hand side to $V_k^s(x) + V_k^{s+1}(x+1)$. Server $x+2$ leads on the left-hand side to $V_k^s(x+1) + V_k^{s+1}(x+1)$, on the right-hand side to $V_k^s(x) + V_k^{s+1}(x+1)$. Both left-hand sides summed are smaller than the right-hand sides summed because of Equation (4), which holds by induction. Next assume that $s \leqslant x$. Then all servers are busy, and the $s+1$th server again gives Equation (4).

Finally, consider the abandonments. Assume that $s \leqslant x$; otherwise there are no abandonments. Note that $V_k^s(x+1)$ and $V_k^{s+1}(x+2)$ have one more customer in queue than $V_k^{s+1}(x+1)$ and $V_k^s(x)$. We can combine the abandonment terms for the first $x-s$ customers in queue, leading again to (2), but with one customer less. The terms concerning the abandonment of the extra customer in queue in $V_k^s(x+1)$ and $V_k^{s+1}(x+2)$ lead to an equality. □

The result as given here is valid only if $\mu \geqslant \gamma$. This leads to two interesting questions. The first is: Are there counterexamples to the result in case $\mu < \gamma$? If this would be the case, then the algorithm could lead to nonoptimal solutions. A second interesting question is the following: is the value function still concave if $\mu < \gamma$? The answer to the second question is no: we will give a counterexample in the next paragraph. This shows that if the result is true in general, then another line of proof would be needed. We did extensive numerical experimentations to try to find a counterexample to the first question. On the basis of this we conjecture that Theorem 2.1 holds for all values of μ and γ.

The simplest counterexample to the concavity of the value function V_k^s is as follows. Take $\lambda = 0$, $\mu = 1$, $s = 1$, $\gamma = 2$, $c = 1$, $r = 0$. Then $V_1^s(x) = -x$, and the departure term of V_2^s is convex. This, summed with a linear cost term, makes V_2^s convex. It is easy to find numerical examples with $\lambda > 0$. As an example, take $\lambda = 15$, $\mu = 1$, $s = 10$, $\gamma = 3$, $c = 0.2$, and $r = 0.1$. Then the direct rewards are strictly concave, but numerical experiments show that the average reward value function is strictly convex. In fact, it can be shown that in the case of linear (or, in general, convex) rewards and a convex total departure rate (in our case, service completions plus abandonments), the value function is convex in the state (Bekker 2008). This does not apply to our situation: the direct rewards are strictly concave due to the rewards for service completions.

Acknowledgments

The authors thank one of the referees for signaling an error concerning concavity of the value function in an earlier version of the proof.

References

Akşin, O. Z., P. T. Harker. 2003. Capacity sizing in the presence of a common shared resource: Staffing an inbound call center. *Eur. J. Oper. Res.* **147**(3) 464–483.

Andrews, B., H. Parsons. 1989. L. L. Bean chooses a telephone agent scheduling system. *Interfaces* **19**(6) 1–9.

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

Armony, M., E. Plambeck, S. Seshadri. 2009. Sensitivity of optimal capacity to customer impatience in an unobservable M/M/S queue (why you shouldn't shout at the DMV). *Manufacturing Service Oper. Management* **11**(1) 19–32.

Bekker, R. 2008. Personal communication.

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

Helber, S., R. Stolletz. 2003. *Call Center Management in der Praxis*. Springer, Berlin.

Helber, S., R. Stolletz, S. Bothe. 2005. Erfolgszielorientierte Agentenallokation in inbound call Centern. (English translation: Goal oriented agent allocation in inbound call centers.) *Zeitschrift für Betriebswirtschaftliche Forschung* (February) 3–32.

Koole, G. M. 1998. Structural results for the control of queueing systems using event-based dynamic programming. *Queueing Systems* **30**(3–4) 323–339.

Koole, G. M. 2006. Monotonicity in Markov reward and decision chains: Theory and applications. *Foundations Trends in Stochastic Systems* **1**(1) 1–76.

Lippman, S. A. 1975. Applying a new device in the optimization of exponential queueing systems. *Oper. Res.* **23**(4) 687–710.

Puterman, M. L. 1994. *Markov Decision Processes*. John Wiley & Sons, New York.

Varga, Á. 2002. Monotonicity properties of single server queues. Master's thesis, Vrije Universiteit, Amsterdam.