Profit maximization for inbound call centres

Sierag, D.D.

D.D. Sierag
dsierag@math.leidenuniv.nl

Profit maximization for inbound call centres

Bachelor thesis, January 17, 2011

Thesis advisor: Dr. F.M. Spieksma

Mathematisch Instituut, Universiteit Leiden

Contents

1 Introduction

'Hoe meer je weet, hoe minder je weet.'
'The more you know, the less you know.'

A. van Polanen.

Driving through endless forests in Swedish Lappland with the Inlandsbanan; a fourteen and a half hour journey from Östersund to Gällivare. Well, there's no more to see here but for reindeers, trees and some small villages allong this 746 kilometer train route, right? Wrong! The staff of Inlandsbanan is quite proud to tell you through the intercom that the train is passing the largest call centre of Sweden. When you look in the way they tell you to, you see a huge building hidden behind uncountable yet finitely many large trees. As the train passes by, you wonder where the people who work here come from; what long distance they must travel to get to work daily. But most of all, as a mathematician, you wonder how to represent a call centre in a mathematical model, and how to optimize it.

This thesis is all about inbound call centres. An inbound call centre is a call centre with incoming calls, e.g., a telephone customer service of a company. First of all, in Section 2, we will explain the way we did research on this project. In Section 3 we will describe a model in detail for inbound call centres, on the basis of the article by G. Koole and A. Pot [3], who did some decent research on profit maximization on inbound call centres. This model deals with incoming calls which arrive at a finite queue, a finite number of agents which serve costumers at a certain rate and the abandonment of impatient costumers in the queue. This model is quite accurate: at a certain rate customers will call a call centre. At this call centre agents are stationed to answer these calls, but sometimes all agents will be busy. If the call centre has some available waiting lines, the customer will be placed in this queue. If the waiting time is quite long, the impatient client may decide to hang up. In Section 4 the expected profit for this model is evaluated in a nice fast way. Using this efficient formula, we describe an algorithm for optimizing the call centre. However, the model is not close enough to reality. Abandoning clients may really need the call, and try to call back later. In Section 5 we introduce a new improved model, which deals with these retrials. Unfortunately, this model is much more complicated than the first model, and there is no efficient algorithm to evaluate the expected profit of the call centre. By approximating the actual value using dynamic programming, we do get some ideas about the structure of the problem. These experiments give very important results, which actually imply that the first, simple and computable model gives results which are far from the optimal values for the actual problem. This will be explained more in detail, when both models are introduced. The conclusion therefor is that determining how to operate the call centre using the first model is easy, but doesn't give you the optimal planning, but determining how to operate the call centre using the second and more reliable model takes too much computation time and therefor is not usefull. More research needs to be done to come up with a better algorithm for determing the best planning for your call centre.

2 Working methods

The subject of this thesis follows from an article written by G. Koole and A. Pot [3]. We studied this article and the corresponding background theory (see G. Koole [2] and L.C.M. Kallenberg [1], page 117-150). As mentioned earlier, the model described by G. Koole and A. Pot [3] has some flaws. The investigation of the retrial model though is much more complicated and therefor we used an approximation algorithm based on dynamic programming using value evaluation (L.C.M. Kallenberg [1], page 192) to get more knowledge about this problem. With this algorithm we did some experiments. Unfortunately, getting results was quite time consuming, and therefor our results are not as expanded as we wished. We had to choose carefully the input data to make sure we really got the data we wanted. With every new dataset, new ideas and conjectures rised, which led to new experiments to verify these conjectures. A nice set of conjectures is given at the end of Section 5.

3 Model description

In this section we will describe the model, to be studied later on. We will consider a model with one waiting line with a finite number of available lines N, with a Poisson arrival rate of λ. We have a finite number S of available agents, who serve customers at an exponential service rate μ for every agent. Furthermore, the first client that arrives in the waiting line is the first to be served, except when he choses to leave beforehand. The latter happens at an exponential rate γ for all clients in the queue. The successive service times, inter arrival times and impatience times are independent of each other. The maximum number of clients in the system is $S + N$ and there are infinitely many clients in the client source. Because the number of clients in the system is limited, the number of clients in the queue is limited and it is possible that arriving customers are rejected from entering. The clients who leave the line at rate γ don't join the group of clients who may call at the same rate, so they will never call again. The number of agents available S is yet undetermined but fixed, with an upper bound of $S_{\max}$. Every agent on duty costs 1 per unit of time. For every handled call, we expect to get a reward of r (for now this expected value is all we are interested in). The number of available waiting lines is also variable but has no upper bound. By introducing costs c per used line per unit of time, it may be more efficient to reduce the number of lines available. If $c = 0$, why bother putting a limit on the number of available lines?

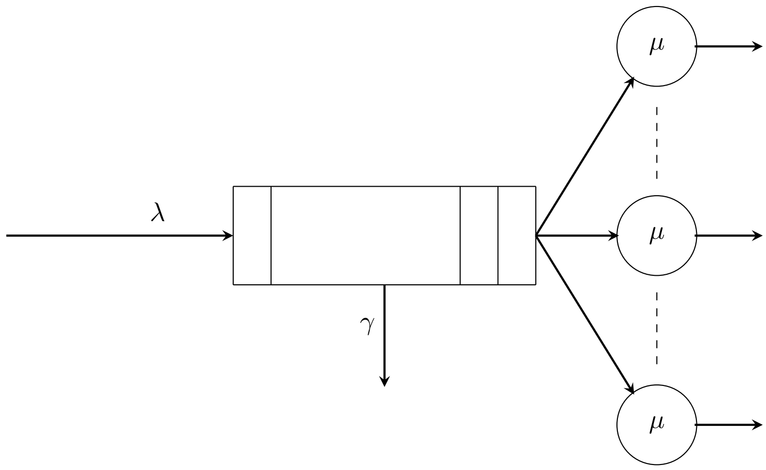

Figure 1: Visualisation of the model.

See Figure 1 for a visualisation of the model. Below we give a summary of the variables and constants:

$$
\begin{aligned}
S_{\max} &= \text{maximum number of available agents,}\\
N_{\max} &= \text{maximum number of available waiting lines,}\\
S &= \text{ number of available agents,}\\
N &= \text{ number of available waiting lines,}\\
x &= \text{ number of clients in the system} \,(\leq S+N)\,,\\
\lambda &= \text{ client arrival rate,}\\
\mu &= \text{ service time rate,}\\
\gamma &= \text{ rate at which clients leave the waiting line,}\\
c &= \text{ costs per line in use per unit of time,}\\
1 &= \text{ costs per agent on duty per unit of time,}\\
r &= \text{ reward per handled call.}
\end{aligned}
$$

We can consider this model as a continuous time Markov process with state space

$$X = \{x | x = \text{ number of clients in the system}\}$$

and transitition rates according to the model discribed above. This is a birth-death process, see Figure 2 for a visualition of the birth-death process.

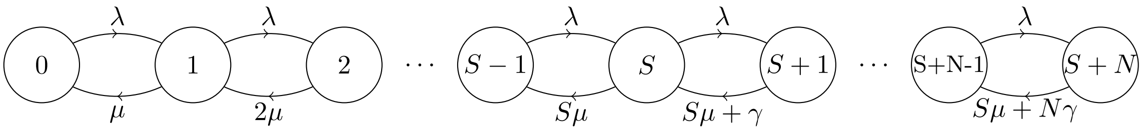

Figure 2: Visualisation of the continuous and discrete Markov Chain.

From this we can derive the generator matrix Q:

$$
Q = \begin{pmatrix}
-\lambda & \lambda & 0 & 0 & . & 0 & . & 0 \\
\mu & -\lambda-\mu & \lambda & 0 & . & 0 & . & 0 \\
0 & 2\mu & -\lambda-2\mu & \lambda & . & 0 & . & 0 \\
. & . & . & . & \ddots & . & & . \\
0 & 0 & 0 & S\mu & -\lambda-S\mu & \lambda & . & 0 \\
. & . & . & . & \ddots & \ddots & \ddots & . \\
0 & 0 & 0 & 0 & . & S\mu+N\gamma & -S\mu-N\gamma & 0 \\
\end{pmatrix}
$$

Remark: the last column can be deleted. It is put here to show that there is no possibility to go to a higher state than state $S+N$ in this birth-death process.

4 Expected profit

We are interested in the expected profit of the call centre described by the model. In this section we will define a function for average long-run profits and give a formula which describes this function for our model. Though this is not clear at first, we will see that this function has some nice properties. These are described in Theorem 4.1 with an extensive proof. With this theorem, we will formulate an effective algorithm for finding optimal values for S and N, given λ, μ, γ, r and c.

4.1 Average long-run profit function

To find a useful profit function, we use uniformization of the Markov Chain. The arrival, service and impatience times are exponentially distributed with parameters λ, γ and μ respectively. Let h be one unit of time (here we choose $h = (\lambda + S_{\max}\mu + N\gamma)^{-1}$ for convenience). Applying uniformization gives us a discrete time Markov Chain. Now the transition rates are probabilities and the discrete Markov Chain can be visualised in the same way we visualised the continuous Markov Chain (see Figure 2).

We are interested in the long-run expected profit. We define $g^{S,N}$ as the average long-run profit for S agents and N waiting lines. For fixed S and N $g^{S,N}$ is given by

$$g^{S,N} = \sum_{x=0}^{S+N} \pi_x \delta_x,$$

with $\delta_x = \min\{x, S\}\mu r - xc - S$, the reward function.

Because the model is an irreducible birth-death process, we can easily calculate the stationary distribution π:

$$\begin{aligned}
\lambda\pi_0 &= \mu\pi_1 &\rightarrow\quad \pi_1 &= \frac{\lambda}{\mu}\pi_0,\\
\lambda\pi_1 &= 2\mu\pi_2 &\rightarrow\quad \pi_2 &= \frac{\lambda}{2\mu}\pi_1 = \frac{\lambda^2}{2\mu^2}\pi_0,\\
&&\vdots&\\
\lambda\pi_{x-1} &= x\mu\pi_x &\rightarrow\quad \pi_x &= \left(\frac{\lambda}{\mu}\right)^x \frac{1}{x!}\pi_0,\\
&&\vdots&\\
\lambda\pi_S &= (S\mu+\gamma)\pi_{S+1} &\rightarrow\quad \pi_{S+1} &= \left(\frac{\lambda}{\mu}\right)^S \frac{1}{S!}\frac{\lambda}{S\mu+\gamma}\pi_0,\\
&&\vdots&\\
\lambda\pi_{S+x-1} &= (S\mu+N\gamma)\pi_{S+x} &\rightarrow\quad \pi_{S+x} &= \left(\frac{\lambda}{\mu}\right)^S \frac{1}{S!}\frac{\lambda^x}{\prod_{y=1}^{x}(S\mu+y\gamma)}\pi_0,\\
&&\vdots&\\
\lambda\pi_{S+N-1} &= (S\mu+N\gamma)\pi_{S+N} &\rightarrow\quad \pi_{S+N} &= \left(\frac{\lambda}{\mu}\right)^S \frac{1}{S!}\frac{\lambda^N}{\prod_{y=1}^{N}(S\mu+y\gamma)}\pi_0.
\end{aligned}$$

And because $\sum_{x=0}^{S+N}\pi_x = 1$, we find

$$\left(1+\frac{\lambda}{\mu}+\ldots+\left(\frac{\lambda}{\mu}\right)^S\frac{1}{S!}\frac{\lambda^N}{\prod_{y=1}^{N}(S\mu+y\gamma)}\right)\pi_0 = 1 \rightarrow \quad \pi_0 = \frac{1}{\sum_{x=0}^{S}\left(\frac{\lambda}{\mu}\right)^x\frac{1}{x!}+\left(\frac{\lambda}{\mu}\right)^S\frac{1}{S!}\sum_{x=1}^{N}\frac{\lambda^x}{\prod_{y=1}^{x}(S\mu+y\gamma)}}.$$

Now can calculate $g^{S,N}$:

$$\begin{aligned}
g^{S,N} &= \sum_{x=0}^{S+N}\pi_x\delta_x,\\
&= \frac{\sum_{x=0}^{S}\left(\frac{\lambda}{\mu}\right)^x\frac{1}{x!}(x\mu r - xc - S)+\left(\frac{\lambda}{\mu}\right)^S\frac{1}{S!}\sum_{x=1}^{N}\frac{\lambda^x}{\prod_{y=1}^{x}(S\mu+y\gamma)}(S\mu r - Sc - xc - S)}{\sum_{x=0}^{S}\left(\frac{\lambda}{\mu}\right)^x\frac{1}{x!}+\left(\frac{\lambda}{\mu}\right)^S\frac{1}{S!}\sum_{x=1}^{N}\frac{\lambda^x}{\prod_{y=1}^{x}(S\mu+y\gamma)}},\\
&= \frac{\sum_{x=0}^{S}\left(\frac{\lambda}{\mu}\right)^x\frac{1}{x!}(x\mu r - xc)+\left(\frac{\lambda}{\mu}\right)^S\frac{1}{S!}\sum_{x=1}^{N}\frac{\lambda^x}{\prod_{y=1}^{x}(S\mu+y\gamma)}(S\mu r - Sc - xc)}{\sum_{x=0}^{S}\left(\frac{\lambda}{\mu}\right)^x\frac{1}{x!}+\left(\frac{\lambda}{\mu}\right)^S\frac{1}{S!}\sum_{x=1}^{N}\frac{\lambda^x}{\prod_{y=1}^{x}(S\mu+y\gamma)}} - S.
\end{aligned} \tag{1}$$

4.2 Properties of the average long-run profit function

The function described above is a very complicated function, and it is not obvious what the values S and N should be that maximise $g^{S,N}$. By making S large, more clients will be served, but the costs will be higher as well. Even so, if we make N large, most lines of N may not be used, so again we spend money for nothing. We should find a good balance between N and S. First we define g^S and g as

$$\begin{aligned}
g^S &= \max_{0\le N\le N_{\max}} g^{S,N}, &\text{with}\quad N_S &= \arg\max_{0\le N\le N_{\max}} g^{S,N},\\
g &= \max_{0\le S\le S_{\max}} g^S, &\text{with}\quad S^* &= \arg\max_{0\le S\le S_{\max}} g^S.
\end{aligned}$$

With this information we can formulate and prove the following theorem.

Theorem 4.1.

1. $g^{S,0} \leq \ldots \leq g^{S,N_S}$ for all $0 \leq S \leq S_{\max}$,
2. $N_S \leq N_{S+1}$ for $\gamma \leq \mu$ and for all $0 \leq S \leq S_{\max}$.

Proof.

1. Assume $N_S > 0$, because else there's nothing to prove. Consider two systems, the first with N available waiting lines and the other with $N+1$ available waiting lines, but both with S available agents. We will compare their stationary distributions. Define the stationary distribution of the first system by π and the stationary distribution of the second one by π'. Because both distributions are stationary and all states up to $S+N$ are the same in both systems, $\frac{\pi_x}{\pi_0} = \frac{\pi'_x}{\pi'_0}$ holds for all x up to $S+N$. Now we can calculate π' by

$$\begin{aligned} \pi'_x &= p\pi_x, \\ \pi'_{S+N+1} &= 1-p, \end{aligned}$$

for all x up to $S+N$ and $p = \frac{\pi'_0}{\pi_0}$. Hence

$$\begin{aligned} g^{S,N+1} &= \sum_{x=0}^{S+N+1} \pi'_x \delta(x), \\ &= \sum_{x=0}^{S+N} p\pi_x \delta(x) + (1-p)\delta(S+N+1) \\ &= pg^{S,N} + (1-p)\delta(S+N+1). \end{aligned}$$

Suppose $g^{S,n} > g^{S,n+1}$ for some $n < N_S$, then $\delta(S+n+1) < g^{S,n}$. The rate is decreasing in x, so for the reward functions we find $\delta(S+N_S) < \ldots < \delta(S+n+1)$. g^{S,N_S} is a convex combination of $\delta(S+N_S), \ldots, \delta(S+n+1)$ and $g^{S,n}$, which means $g^{S,n} > g^{S,N_S}$. This, by iterating, is in contradiction with the optimality of N_S.

2. To prove this assertion, we will use dynamic programming. We formulate the dynamic programming value function $V_k^S(x)$, with k the number of the iteration step, S the number of available agents and x the state, as follows. In each iteration of the value function, we will decide whether the optimal action will be admission or rejection, when a new client arrives. This under the assumption that we keep the number of available agents S constant. Remember we saw earlier that the transition rates in the discrete model can be seen as probabilities. Suppose in iteration step k we are in state x, with S available agents. There are several components in the value function: there are certain costs per iteration; at rate λ a new client arrives, so then we should decide whether to allow or reject the client; at a certain rate clients leave the system; at a certain rate we stay in the same state. First we implement the rewards. When an arriving client is allowed to enter, we receive a reward of r. When a client decides to leave the waiting line, it will cost us r. In state x this occurs at rate $\gamma\,(x-S)^+$. The costs for the lines in use are xc, and the costs for the agents is S. So the reward component is

$$-\gamma r\,(x-S)^+ - cx - S.$$

At rate λ we decide whether we should allow or reject the arriving client, so we should decide whether $V_k^S(x)$ or $r + V_k^S(x+1)$ is the maximum of both. Hence this component is

$$\lambda \max\left\{V_k^S(x), r + V_k^S(x+1)\right\}.$$

At rate $\mu \min\{S,x\}$ a client is served, so at rate $\mu \min\{S,x\} + (\gamma\,(x-S))^+$ we end up in state $x-1$ in the next iteration step. Hence this component is

$$\left[\mu \min\{S,x\} + \gamma\,(x-S)^+\right] V_k^S(x-1).$$

We stay in the same state, if no call ends during that iteration step and no new client arrives. The last one is handled in the component with rate λ. The other one occurs in two different occasions. First, at rate μ an agent who doesn't serve a client should be considered. There are $S_{\max} - \min\{S, x\}$ agents like that in state x. Second, the available waiting lines where no client is waiting shoud be considered. This occurs at $N_{\max} - (x - S)^+$ lines. Hence this component is

$$\left[\mu\left(S_{\max} - \min\{S, x\}\right) + \gamma\left(N_{\max} - (x - S)^+\right)\right] V_k^S(x).$$

Combining these components we find

$$\begin{aligned} V_{k+1}^S(x) = &- \gamma r\,(x - S)^+ - cx - S + \lambda \max\left\{V_k^S(x), r + V_k^S(x+1)\right\} + \\ &\left[\mu \min\{S, x\} + \gamma\,(x - S)^+\right] V_k^S\,(x - 1) + \\ &\left[\mu\left(S_{\max} - \min\{S, x\}\right) + \gamma\left(N_{\max} - (x - S)^+\right)\right] V_k^S(x), \end{aligned} \tag{2}$$

for $x < S + N$ and $k > 0$, and

$$\begin{aligned} V_{k+1}^S(S+N) = &- \gamma r N - c(S+N) - S + \lambda V_k^S(S+N) + \left[\mu S + \gamma N\right] V_k^S\,(S+N) + \\ &\left[\mu\left(S_{\max} - S\right)\right] V_k^S(S+N), \end{aligned}$$

Now suppose a total number of N_S available waiting lines is optimal for a total number of S agents. This means, that if at some time there are x clients in the system and another client arrives, he will be admitted to the waiting line if $x < S + N_S$ and rejected if $x = N_S$. So to prove $N_S \le N_{S+1}$ we need to prove the following: when admission is optimal for x clients in the system with S available agents, then admission is also optimal for $x + 1$ clients in the system with $S + 1$ servers. The following statement will simplify our proof. If

$$V_k^S(x+1) + V_k^{S+1}(x+1) \le V_k^S(x) + V_k^{S+1}(x+2) \tag{3}$$

for all $k > 0$, $0 \le S < S_{\max}$ and $0 \le x < S + N - 1$, then admission is optimal for the system with $x + 1$ clients and $S + 1$ agents if admission is optimal for the system with x clients and S agents. To prove this statement, suppose admission is optimal in the system with x clients and S agents. Then from equation (2) we find $V_k^S(x) \le r + V_k^S(x+1)$. Summing this equation with equation (3) we get $V_k^S(x+1) \le r + V_k^S + 1(x+2)$, so admission is also optimal in the system with $x + 1$ clients and $S + 1$ agents. From Markov decision theory it now follows that the same holds for the long-run limiting average case.

If $\gamma \le \mu$, which is the case by assumption, concavity of $V_k^S(x)$ holds:

$$V_k^S(x) + V_k^S(x+2) \le 2V_k^S(x+1). \tag{4}$$

We need concavity later on in the proof. Summing equation (3) and (4), with S replaced by $S + 1$ in the latter, gives us supermodularity:

$$V_k^S(x+1) + V_k^{S+1}(x) \le V_k^S(x) + V_k^{S+1}(x+1). \tag{5}$$

Now we will prove equation (3), by induction to k. For $k = 0$, equation (3) is trivial.

Now suppose (3) holds up to some k. Now consider the corresponding four terms in inequality (3) with k replaced by $k + 1$. First we consider the reward term,

$$\begin{aligned} &- \gamma r\,(x + 1 - S)^+ - c(x+1) - S - \gamma r\,(x - S)^+ - c(x+1) - (S+1) \\ &\le -\gamma r\,(x - S)^+ - cx - S - \gamma r\,(x + 1 - S)^+ - c(x+2) - (S+1), \end{aligned}$$

which actually is an equality.
Next consider the term with coefficient λ. We need to look at three different cases. In the first case, suppose that the maximizing action is admission in both $V_k^S(x+1)$ and $V_k^{S+1}(x+1)$. Then

$$\begin{aligned}
&\max\left\{V_k^S(x+1), r+V_k^S(x+2)\right\}+\max\left\{V_k^{S+1}(x+1), r+V_k^{S+1}(x+2)\right\}\\
&\quad= \ 2r+V_k^S(x+2)+V_k^{S+1}(x+2)\\
&\quad\leq \ 2r+V_k^S(x+1)+V_k^{S+1}(x+3)\\
&\quad\leq \ \max\left\{V_k^S(x), r+V_k^S(x+1)\right\}+\max\left\{V_k^{S+1}(x+2), r+V_k^{S+1}(x+3)\right\}
\end{aligned}$$

The proof of the second case, where we suppose that in both $V_k^S(x+1)$ and $V_k^{S+1}(x+1$ rejection is optimal, has an analogous proof.
In the third case we suppose that the optimal actions are different. Here admission is the optimal action in $V_k^{S+1}(x+1)$, by induction. Then

$$\begin{aligned}
&\max\left\{V_k^S(x+1), r+V_k^S(x+2)\right\}+\max\left\{V_k^{S+1}(x+1), r+V_k^{S+1}(x+2)\right\}\\
&\quad= \ r+V_k^S(x+1)+V_k^{S+1}(x+2)\\
&\quad\leq \ \max\left\{V_k^S(x), r+V_k^S(x+1)\right\}+\max\left\{V_k^{S+1}(x+2), r+V_k^{S+1}(x+3)\right\}.
\end{aligned}$$

Here the induction hypothesis is used in the first inequality. Now consider the terms with coefficient μ. First we assume that $S \geq x+1$. Then there are no clients in the waiting line, and we must prove

$$\begin{aligned}
&(x+1)V_k^S(x)+(S_{\max}-x-1)V_k^S(x+1)+(x+1)V_k^{S+1}(x)+(S_{\max}-x-1)V_k^{S+1}(x+1)\\
&\quad\leq \ xV_k^S(x-1)+(S_{\max}-x)V_k^S(x)+(x+2)V_k^{S+1}(x+1)+(S_{\max}-x-2)V_k^{S+1}(x+2).
\end{aligned}$$

This is equivalent to

$$\begin{aligned}
&xV_k^S(x)+V_k^S(x)+(S_{\max}-x-2)V_k^S(x+1)+V_k^S(x+1)\\
&\qquad+xV_k^{S+1}(x)+V_k^{S+1}(x)+(S_{\max}-x-2)V_k^{S+1}(x+1)+V_k^{S+1}(x+1)\\
&\quad\leq \ xV_k^S(x-1)+(S_{\max}-x-2)V_k^S(x)+2V_k^S(x)\\
&\qquad+xV_k^{S+1}(x+1)+2V_k^{S+1}(x+1)+(S_{\max}-x-2)V_k^{S+1}(x+2).
\end{aligned}$$

We will split this inequality into three components and prove them:

(a) $xV_k^S(x)+xV_k^{S+1}(x) \overset{?}{\leq} xV_k^S(x-1)x+V_k^{S+1}(x+1).$

By multiplying equation (3) with x we get the equation above. (3) holds by induction.

(b) $(S_{\max}-x-2)V_k^S(x+1)+(S_{\max}-x-2)V_k^{S+1}(x+1) \overset{?}{\leq} (S_{\max}-x-2)V_k^S(x)+(S_{\max}-x-2)V_k^{S+1}(x+2).$

This equation we get by multiplying (3) with $S_{\max}-x-2$.

(c) $V_k^S(x)+V_k^S(x+1)+V_k^{S+1}(x)+V_k^{S+1}(x+1) \overset{?}{\leq} 2V_k^S(x)+2V_k^{S+1}(x+1).$

Rewriting this equation gives us equation (5), which holds by induction.

Now assume $S \leq x$. Now we must prove

$$\begin{aligned}
&SV_k^S(x)+(S_{\max}-S)V_k^S(x+1)+(S+1)V_k^{S+1}(x)+(S_{\max}-S-1)V_k^{S+1}(x+1)\\
&\quad\leq \ SV_k^S(x-1)+(S_{\max}-S)V_k^S(x)+(S+1)V_k^{S+1}(x+1)+(S_{\max}-S-1)V_k^{S+1}(x+2),
\end{aligned}$$

Using simular arguments as above we find this statement to be true.

Now consider the final terms, with coefficient γ. We must prove

$$\begin{array}{rr} & (x+1-S)^+ V_k^S(x) + (N_{\max} - (x+1-S)^+) V_k^S(x+1) \\ & +(x-S)^+ V_k^{S+1}(x) + (N_{\max} - (x-S)^+) V_k^{S+1}(x+1) \\ \leq & (x-S)^+ V_k^S(x-1) + (N_{\max} - (x-S)^+) V_k^S(x) \\ & +(x+1-S)^+ V_k^{S+1}(x+1) + (N_{\max} - (x+1-S)^+) V_k^{S+1}(x+2). \end{array} \quad (6)$$

We consider two possibilities:

(a) Suppose $x+1 \leq S$. Then inequality(6) is equal to $N_{\max}$ times equation (3), which holds by induction.

(b) Suppose $x \geq S$. Now we obtain inequality (6) by multiplying equation (3) by $((x-S)+(N+S-x-1))$ and summing this with the equality

$$V_k^S(x) + V_k^{S+1} = V_k^S(x) + V_k^{S+1}.$$

□

4.3 Algorithm for finding (S^*, N_{S^*})

Theorem 4.1 leads to an effective algorithm, so we don't have to calculate $g^{S,N}$ for all S, N. According to Theorem 4.1, the function $g^{S,N}$ of N with fixed S is increasing until N_S. In fact, N_S gives the highest value for $g^{S,N}$ for fixed S. We thus just need to calculate $g^{S,N}$ for $0 \leq N \leq N_S + 1$, with $g^{S,N_S+1} < g^{S,N_S}$. Combining this conclusion with theorem 2.1.1 and supposing that we know g^{S,N_S}, we see that we just need to calculate $g^{S+1,N}$ with $S+1$ fixed for $N_S \leq N \leq N_{S+1}+1$, with $g^{S+1,N_{S+1}+1} < g^{S+1,N_{S+1}}$.
In the algorithm we first check whether $0 \geq r\mu - c - 1$ holds. If this is the case, it's better to reject all calls, because the costs per agent and the comminucation costs of the call are higher than the profits. Below we give the algorithm:

Algorithm 4.1. *Algorithm for finding* (S^*, N_{S^*})

1. If $0 \geq r\mu - c - 1$, then STOP.
2. Take $(S, N) = (S^*, N_{S^*}) = (0, 0)$.
3. For $S = 1$ to $S_{\max}$ do
   (a) Compute $g^{S,N}$ (using equation 1).
   (b) If $N < N_{\max}$, then compute $g^{S,N+1}$.
   (c) While $g^{S,N} < g^{S,\min\{N+1,N_{\max}\}}$
      i. $N = N + 1$.
      ii. If $N < N_{\max}$, then compute $g^{S,N+1}$.
   (d) If $g^{S,N} > g^{S^*,N_{S^*}}$, then $(S^*, N_{S^*}) = (S, N)$.

Concluding this section we will derive the complexity of Algorithm 4.1. Clearly, step 1 and 2 are of order 1. Step 3 is more complicated. First of all, we have $S_{\max}$ iteration steps. Because of Theorem 4.1 we don't have to look at all $N \in \{1, \cdots, N_{\max}\}$ in each iteration step. We will describe the worst case scenario. We assume that $N_{\max}$ is not the optimal solution for $S < S_{\max}$, but it is the optimal solution for $S_{\max}$. In the first iteration we have to evaluate $g^{1,N}$ for $N \in \{1, \cdots, N_1, N_1+1\}$. In iteration S with $S \in \{2, \cdots, S_{\max}\}$ we evaluate $g^{S,N}$ for $N \in \{N_{S-1}, \cdots, N_S, N_S+1\}$. In the final iteration step $S_{\max}$ we evaluate $g^{S_{\max},N}$ for $N \in \{N_{S_{\max}-1}, \cdots, N_{S_{\max}} = N_{\max}\}$. We see we have to evaluate and look at all $N \in \{1, \cdots, N_{\max}\}$. But we have to evaluate N_S and $N_S + 1$ twice for all $S \in \{1, \cdots, S_{\max} - 1\}$. Therefor the complexity of step 3 is of order $N_{\max} + 2S_{\max}$.

5 Retrial model

In the model discribed above, we assumed the abandonments wouldn't return to the queue. Now we will look at a model where the abonments will be kept in a pool, as if they were impatient but really need the call, so they will try and call again later to join the queue. In the real world it is very realistic that people will hang up before getting served and call back later, so it could be wise to investigate this problem. First we will describe the model in more detail and we point out some problems which prevent this problem from being as easy to solve as the model described earlier. Next we give a method to approximate the value of the long-run profits for S agents and N available waiting lines using dynamic programming. With this method we show some experiments on the new model, with some fascinating results. It turnes out that the more realistic, retrial model, provides results that differ very much from the first model. We provide some conjectures, derived from the experiments, concerning the difference between the models.

5.1 Model description

In this model we assume that the abandonments stay in a pool with capacity Y, waiting to return to the queue at rate β. See Figure 3 for a visualisation of the model. Analyzing this model is much more complicated, even though it's slightly different from the first one. Not only do we have to keep track of the number of clients in the system, we also need to keep track of the number of clients in the pool. This gives a two dimensional state space, so, although we're dealing with a Markov Chain, it isn't a birth-death process anymore and we lose all it's nice properties.

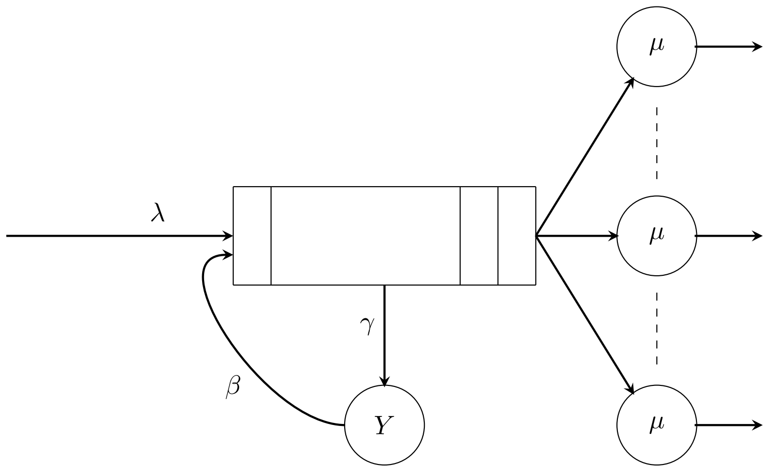

Figure 3: Visualisation of the retrial model.

The transition rates have changed, so we need to reconsider them. As before we define $N_{\max}$ to be the maximum number of available waiting lines and $S_{\max}$ the maximum number of available agents, N the number of available waiting lines and S the number of available agents. Clients still arrive at rate λ from the infinite client source and clients get served at rate μ by the agents. Impatient clients still leave at rate γ, but now they will be stationed at the retrial pool, which they will leave at rate β. Then they will join the waiting line again, hopefully, for both client and call center manager, with a better ending. We now define x as the number of clients who are in the basic system, i.e., who are in either the waiting line or getting served by an agent. For the retrial pool we define y as the number of clients in the pool. Later on we will need an upper bound for the pool, which we define by Y. Costs and rewards remain the same. Let us list the variables:

$$\begin{aligned}
S_{\max} &= \text{maximum number of available agents,}\\
N_{\max} &= \text{maximum number of available waiting lines,}\\
Y &= \text{upper bound of the retrial pool,}\\
S &= \text{ number of available agents,}\\
N &= \text{ number of available waiting lines,}\\
x &= \text{number of clients in the basic system} \, (\leq S+N) \, ,\\
y &= \text{ number of clients in retrial pool} \, (\leq Y) \, ,\\
\lambda &= \text{ client arrival rate,}\\
\mu &= \text{ service time rate,}\\
\gamma &= \text{ rate at which clients leave the waiting line,}\\
\beta &= \text{ retrial rate.}
\end{aligned}$$

Now we can describe the new transition rates. Suppose (x, y) is our present state, with x the number of clients in the basic system and y in the retrial pool. Because there are just four different transition rates, there are four possible states after the next event:

1. The first one is $(x-1, y)$, which occurs at rate $\min(x, S)\mu$. If a client leaves the system after finishing service, the number of clients in the pool remain unchanged, but the number of clients in the basic system drops by one.
2. The second possibility is $(x+1, y)$, which occurs at rate λ. At this rate clients arrive from the infinite client source to the basic system.
3. The third possibility is $(x-1, y+1)$, which occurs at rate $\max\{x-S, 0\}\gamma$. Now a client leaves the waiting line and the basic system, but he enters the retrial pool.
4. The fourth and final possibility is $(x+1, y-1)$, which occurs at rate $y\beta$. A client leaves the retrial pool and he joins the waiting line again.

See Figure 4 for a visualitation of the transition rates.

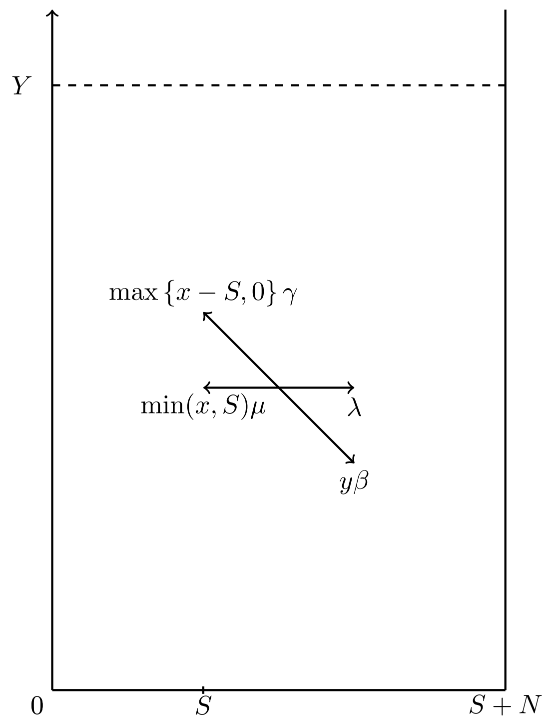

Figure 4: Visualisation of the two dimensional Markov Chain.

We would like to get more knowledge about this new problem. We would like to find out if the retrial model also has nice properties, like the ones described in Theorem 4.1. However there are some obstacles allong our path.

The first problem we run into, is the fact that evaluating the stationary distribution is hard. We find that it is possible to evaluate it for small problems, but even then the formulas are huge. Though we didn't find any nice structure for evaluating the stationary distribution, this is a subject of interest for further research. Without the stationary distribution, it is also hard to evaluate the value function. In the algorithm for the first problem we used this function, assuming it was easy to calculate. Now it is not so easy to evaluate it, we must consider other methods. A well known and appreciated method is value evaluation using dynamic programming.

5.2 Value function

Before we define the value function, we look at the state space. In the description of the model, the retrial pool has infinite capacity. If β is way smaller than γ, the number of clients will become very large and may tend to infinity as time goes to infinity. Unfortunately our computer programs can't deal with this infinite pool. To prevent this problem, we introduce an upper bound Y to the retrial pool. What Y should be, depends on the other parameters. An important part is the relation between β and γ. We will come back to this later, when we describe the program.

We formulate the value function $V_k^{S,N}(x, y)$. With this function we approximate the value of $g^{S,N}$ for fixed S and N. We scale time such that $\lambda + S_{\max}\mu + N_{\max}\gamma + Y\beta = 1$. Notice that this is only possible because Y is finite. We impliment the costs as before: when an arriving customer is allowed to enter, i.e., when $x < S + N$, we receive a reward of r. When a customer abandons we incur a cost of r. In state (x, y) this happens at rate $\gamma \max\{x - S, 0\}$. The costs for communication is $cx + S$ for state (x, y). The reward component is

$$-\gamma r \max\{x - S, 0\} - cx - S.$$

Now we describe the different components of the value function corresponding to the different transition rates. For notational convenience, we will write $V_k(x, y)$ instead of $V_k^{S,N}(x, y)$. First we describe the one with transition rate λ. This component is

$$\lambda(r + V_k(x + 1, y)).$$

At rate $\mu \min\{x, S\}$ a client leaves the whole system by being served. Therefor we end up in state $(x - 1, y)$. Hence this component is

$$\mu \min\{x, S\} V_k(x - 1, y).$$

At rate $\gamma \max\{x - S, 0\}$ a client leaves the waiting line and enters the retrial pool. Hence this component is

$$\gamma \max\{x - S, 0\} V_k(x - 1, y + 1).$$

At rate βy a client enters the waiting line from the retrial pool. Hence this component is

$$\beta y(r + V_k(x + 1, y - 1)).$$

The rate at which we stay in the current state is 1 minus the other transition rates. Hence thiscomponent is

$$(1 - \lambda - \mu \min\{x, S\} - \beta y)V_k(x, y).$$

Combining all components we find

$$\begin{aligned} V_{k+1}(x, y) = &- \gamma r \max\{x - S, 0\} - cx - S + \lambda(r + V_k(x + 1, y)) + \mu \min\{x, S\} V_k(x - 1, y) \\ &+ \gamma \max\{x - S, 0\} V_k(x - 1, y + 1) + \beta y(r + V_k(x + 1, y - 1)) \\ &+ (1 - \lambda - \mu \min\{x, S\} - \beta y)V_k(x, y), \end{aligned}$$

for $x < S + N$ and $y < Y$,

$$\begin{aligned} V_{k+1}(x, y) = &- \gamma r \max\{x - S, 0\} - cx - S + \lambda V_k(x, y) + \mu \min\{x, S\} V_k(x - 1, y) \\ &+ \gamma \max\{x - S, 0\} V_k(x - 1, y + 1) + \beta y V_k(x, y - 1) \\ &+ (1 - \lambda - \mu \min\{x, S\} - \beta y)V_k(x, y), \end{aligned}$$

for $x = S + N$ and $y < Y$,

$$\begin{aligned}V_{k+1}(x,y) = &-\gamma r \max\{x-S,0\} - cx - S + \lambda(r + V_k(x+1,y)) + \mu \min\{x,S\} V_k(x-1,y)\\ &+ \gamma \max\{x-S,0\} V_k(x-1,y) + \beta y(r + V_k(x+1,y-1))\\ &+ (1-\lambda-\mu\min\{x,S\} - \beta y)V_k(x,y),\end{aligned}$$

for $x < S+N$ and $y = Y$,

$$\begin{aligned}V_{k+1}(x,y) = &-\gamma r \max\{x-S,0\} - cx - S + \lambda V_k(x,y) + \mu \min\{x,S\} V_k(x-1,y)\\ &+ \gamma \max\{x-S,0\} V_k(x-1,y) + \beta y V_k(x+1,y-1)\\ &+ (1-\lambda-\mu\min\{x,S\} - \beta y)V_k(x,y),\end{aligned}$$

for $x = S+N$ and $y = Y$.

Though the retrial pool has infinite capacity, our computer doesn't have infinite capacity. Especially when γ is way bigger than β, the retrial pool will become very large. We must choose an upper bound Y to program the retrial pool in such a way that the approximated value for $g^{S,N}$ does not vary too much from $g^{S,N}$. In the algorithm we choose $\epsilon > 0$ for the accuracy of the algorithm, so it's a good idea to let Y depend on ϵ. In the worst case scenario, the transition rate to go from state y to $y+1$ is $N_{\max}\gamma$. Suppose we look just at the retrial pool and forget about the regular system, assuming the worst case scenario. A visualisation of the Markov Chain is given in Figure 5. This is a birth-death process, so we can calculate it's stationary distribution easily. We find

$$\begin{aligned}\pi_y &= \left(\frac{N_{\max}\gamma}{\beta}\right)^y \frac{1}{y!}\pi_0, \qquad\qquad 0 < y \leq Y,\\ \pi_0 &= \frac{1}{\sum_{y=0}^{Y}\left(\frac{N_{\max}\gamma}{\beta}\right)^y \frac{1}{y!}},\\ \pi_Y &= \frac{\left(\frac{N_{\max}\gamma}{\beta}\right)^Y \frac{1}{Y!}}{\sum_{y=0}^{Y}\left(\frac{N_{\max}\gamma}{\beta}\right)^y \frac{1}{y!}} = \frac{1}{\sum_{y=0}^{Y}\left(\frac{\beta}{N_{\max}\gamma}\right)^{Y-y} \frac{Y!}{y!}} = \frac{1}{\sum_{u=0}^{Y}\left(\frac{\beta}{N_{\max}\gamma}\right)^{u} \frac{Y!}{(M-u)!}},\end{aligned}$$

with $Y-y$ replaced by u in the last equation. We say that Y is large enough if π_Y is small enough, i.e., $\pi_Y < \frac{1}{2}\epsilon$. This way we get a $\frac{1}{2}\epsilon$-approximation of Y. This means that almost never a state higher than Y will be reached in the unlimited case; states higher than Y are negligible.
Remark: Y doesn't depend on S or N, so we can use the same Y for the evaluation of $g^{S,N}$ for $1 \leq S \leq S_{\max}$, $0 \leq N \leq N_{\max}$.

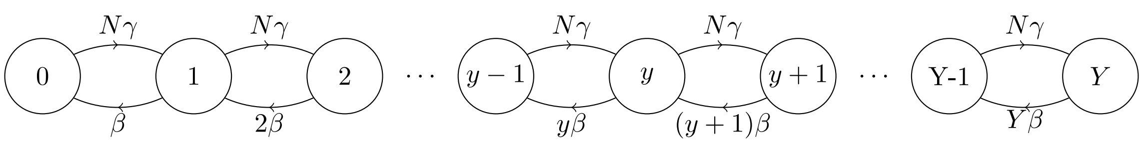

Figure 5: Visualisation of the Markov Chain.

Let

$$X = \{(x,y) | x = \text{number of clients at the servers or in waiting line}, y = \text{number of clients in the retrial pool}\}$$

be the state space of the system. Let p_{ij} be the transition rate to go from state $i \in X$ to state $j \in X$. Let $r_i = -\gamma r \max(0, x-S) - cx - S$ be the reward function of state $i = (x,y) \in X$. To evaluate $g^{S,N}$ we use the following algorithm, derived from a basic value evaluation algorithm (L.C.M. Kallenberg [1], page 192):

Algorithm 5.1. *Value evaluation using dynamic programming*

1. Choose $\epsilon > 0$ and arbitrary $v \in \mathbb{R}^{S+N+1} \times \mathbb{R}^{Y+1}$ with $v_{(S+N+1,Y+1)} = 0$.

2. (a) Compute $z_i = r_i + \sum p_{ij} v_i$, $i \in X$.
   (b) $f = z_{(S+N+1,Y+1)}$.
   (c) $w_i = z_i - f$, $i \in X$.
   (d) $u = \max_{i \in X}(w_i - v_i)$, $l = \min_{i \in X}(w_i - v_i)$.
3. If $u - l \leq \epsilon$: $\frac{1}{2}(u+l) + f$ is a $\frac{1}{2}\epsilon$-approximation of $g^{S,N}$. STOP.
   Else: $v := w$ and proceed with step 2.

5.3 Experiments

In order to formulate new theories, a basis for ideas is needed. Because our problem is quite difficult, we did some experiments to get these ideas. The outcomes are actually quite astonishing and very much the contrary of our predictions. First of all, intuitively, we expected that by introducing a relatively small value of β, the optimal values for S and N and therefor $g^{S,N}$ would not vary that much from the optimal values for the model without retrials. If this is the case, why bother modeling retrials, for this problem is much more difficult, while the solution derived by the model without retrials gives almost perfect values? In fact, for small β, these values fluctuate heavily, and therefor a great amount of rewards is discarded when using the model without retrials.
Also, intuitively, we expected S^* to only decrease in value if abandonments are added, i.e., when we look at the model without retrials. When $\gamma > 0$, we might consider this as a model without abandonments but with a lower value for λ, for less clients will arrive at agents. But this is not the case. In fact, for $\gamma > 0$, S^* takes larger values.

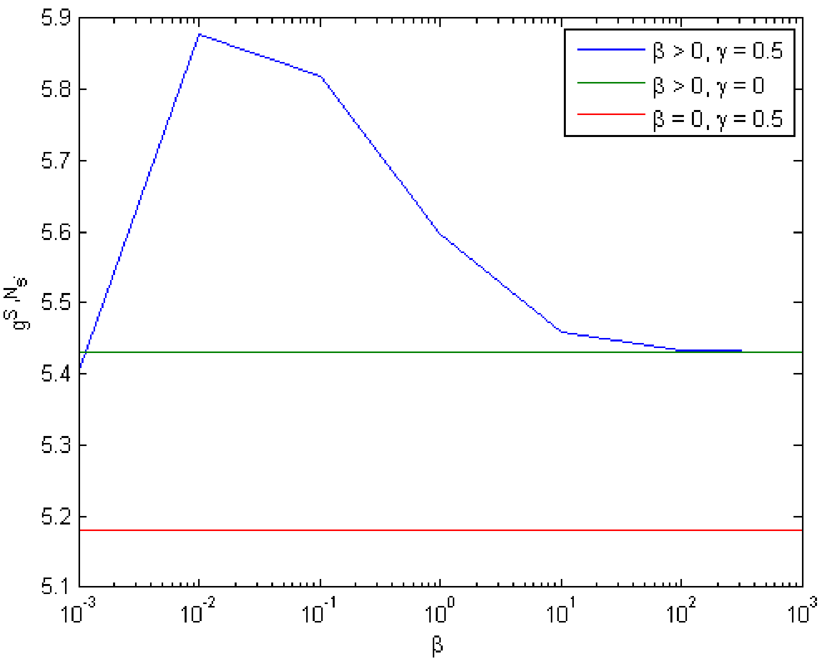

Figure 6: Plot of $g^{S^*,N_{S^*}}$ for 2 graphs for $\beta \in \left\{0, \frac{1}{1000}, \frac{1}{100}, \frac{1}{10}, 1, 10, 100, 1000\right\}$.

In the first experiment we use the following values for the variables:

$$\begin{aligned} S_{\max} &= 8, & \lambda &= 5, \\ N_{\max} &= 30, & \mu &= 1, \\ r &= 3, & \gamma &= 0.5, \\ c &= 0.5, & \epsilon &= 0.001, \end{aligned}$$

with different values for β. In Figure 7 a table of the values of g^{S,N_S} is shown. This experiment is an explicit example that shows there is no monotonicity of $g^{S,N}$ when varying β. A plot corresponding to values of $g^{S^*,N_{S^*}}$ is given in Figure 6. Here we can see clearly that the value $g^{S^*,N_{S^*}}$ really depends on the value of β. In fact, for

	S							
β	1	2	3	4	5	6	7	8
0	1.083	2.310	3.410	4.362	5.003	5.180	4.865	4.215
$\frac{1}{1000}$	1.083	2.326	3.439	4.426	5.135	5.401	5.163	4.409
$\frac{1}{100}$	1.083	2.338	3.493	4.572	5.482	5.877	5.272	4.410
$\frac{1}{10}$	1.083	2.339	3.493	4.567	5.450	5.819	5.262	4.407
1	1.082	2.339	3.484	4.526	5.312	5.598	5.192	4.388
10	1.083	2.340	3.463	4.466	5.197	5.460	5.121	4.366
100	1.083	2.335	3.449	4.444	5.173	5.434	5.107	4.360
1000	1.083	2.334	3.447	4.441	5.170	5.431	5.105	4.360

Figure 7: $\frac{1}{2}\epsilon$-aproximation of g^{S,N_S} with $\lambda = 5$, $\mu = 1$, $\gamma = 0.5$, $r = 3$, $c = 0.5$ and $\epsilon = 0.001$.

small values of β, the value of $g^{S^*,N_{S^*}}$ varies a lot. Since the model with the retrial pool is more plausible than without, it's rather important to take this variation for small β seriously. Using the model without the pool, we loose a profit of almost 12%. We also find this phenomenon for fixed S and varying β. For example, in Figure 7, for $S = 6$ we find an optimal value of 5.877 for $\beta = \frac{1}{100}$, while for $\beta = 0$ we find a value of 5.180. Actually, for all S a higher reward is earned for $\beta > 0$ in comparison with the reward for $\beta = 0$, though the difference is very small for S small enough (and probably also for S large enough; evaluating g^{S,N_S} is rather time consuming for large S).

Also N_{S^*} varies a lot for small values of β, see Figure 8 for an example. Let $N^{\beta}_{S^*}$ be the optimal number of available lines in the queue for the system with $\beta \geq 0$. This value $N^{\beta}_{S^*}$ with $\beta > 0$ is much larger, but never smaller, than the value of $N^0_{S^*}$ evaluated using Algorithm 4.1. We formulate our first conjecture:

Conjecture 5.1. *Let λ, μ and γ be fixed. Then*

$$N^0_{S^*} \leq N^{\beta}_{S^*}$$

for all $\beta > 0$.

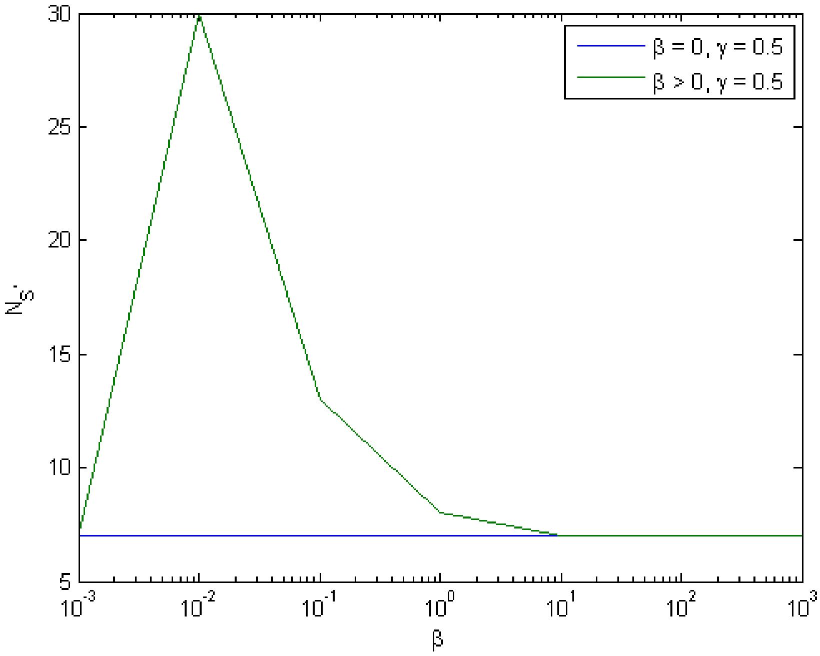

Figure 8: Plot of N_{S^*} against $\beta \in \{0, \frac{1}{1000}, \frac{1}{100}, \frac{1}{10}, 1, 10, 100, 1000\}$ for 2 graphs.

Once again consider Figure 7. Define β_S as the optimal value for β for fixed S, and $g^{S,N}_{\beta}$ as the long run expected profit function for fixed S, N and β. For each $S > 1$ we've underlined the optimal value in the corresponding column, i.e., $g^{S,N}_{\beta_S}$. We see β_S decreasing when S increases. This monotonicity leads to the following conjecture:

Conjecture 5.2. *Let λ, μ and γ be fixed. Then the following statement holds for $S \leq S'$:*

$$\beta_{S'} \leq \beta_S.$$

Because it's hard to get information about the function $g^{S,N}_{\beta}$ for large S, β_S might increase for large S. If this is true, the following conjecture might be of interest:

Conjecture 5.3. *Let λ, μ and γ be fixed. Then the following statement holds for $S \leq S' \leq S^*$:*

$$\beta_{S^*} \leq \beta_{S'} \leq \beta_S.$$

Remark: this conjecture is just a special case of the first one.

In Figure 6 we find another plot, a graph of g^{S,N_S} with the same parameters but for the abandonments, so $\gamma = 0$. We see that for β large, the graph with abandonments tents to the one without. Naturally the model without abandonments is a M/M/S; FIFO/(N-S)/∞ model (see Kallenberg [1], page 130). This observation is not very strange. When β is large, when a client enters the retrial pool, he almost certainly will return directly back to the queue, as if he never left; in other (more mathematical) words: as if $\gamma = 0$. Let $g^{S,N}_{\beta,\gamma}$ be the long run profit function for fixed γ and β (we just add β and γ to show with that β and γ we are working with). For this observation we formulate the following conjecture:

Conjecture 5.4. *Let λ, μ and γ be fixed. Then the following statement holds:*

$$\lim_{\beta\to\infty} g^{S^*,N_{S^*}}_{\beta,\gamma} = g^{S^*,N_{S^*}}_{0,0}.$$

Another interesting observation is the fact that in the example given in Figure 7 S^* remains constant, even though β takes different values. If this is the case more often, we could form a general statement. Why would it be the case that for given λ, μ, γ, r and c, S^* remains constant for all β? As we saw earlier, for $\gamma = 0$ we find a M/M/S; FIFO/(N-S)/∞ model. The evaluation of the stationary distribution is now easily computed with the formula of Little (Kallenberg [1], page 121 and 125). Therefor it is also easy to compute S^*_0. For arbitrary λ, μ and γ this is obviously not the case. For large γ, relative to λ, we might consider this model as a M/M/S; FIFO/(N-S)/∞ model with $\lambda' < \lambda$ as arival rate. For γ large enough, λ' is small enough to provide a lower S^* than the model with abandonments. But we need not to make a generalisation for all γ. In fact, if we look back to the model without the retrial pool: in Theorem 4.1 we assumed $\gamma \leq \mu$. Let us now also assume $\gamma \leq \mu$. Unfortunately, this statement is not enough. In the table in Figure 9 an example is given for which this statement is not true.

	S			
γ	1	2	3	4
0	2.3772727e+001	2.7333333e+001	2.6477273e+001	2.5496599e+001
$\frac{1}{2}$	1.8952424e+001	2.5225353e+001	2.6041411e+001	2.5420104e+001
1	1.7463617e+001	2.4390850e+001	2.5799892e+001	2.5369537e+001

Figure 9: Evaluation of $g^{S^*,N_{S^*}}$ using equation (1), for $\lambda = 1$, $\mu = 1$, $\beta = 0$, $r = 30$, $c = 0.5$. For $\gamma = 0$, $S^* = 2$, while for $\gamma = 1$ and $\gamma = 2$, $S^* = 3$.

This example gives another remarkable insight. When parameter γ is introduced, this leads to a more slow flow of clients, since they leave the queue, and therefor we could interpret this model just by using a lower value of λ. If we use a lower value for λ, S^* might take lower values, but never higher values. In this example, however, we see that S^* takes higher values for $\gamma > 0$. This holds for the model without retrials. Let $S^*_{\beta,\gamma}$ be the optimal value for S corresponding to our model with $\beta, \gamma \geq 0$. We form the following conjecture:

Conjecture 5.5. *Let $\beta > 0$, $0 < \gamma \leq \mu$. Then $S^*_{\beta,\gamma}$ will take values*

$$S^*_{0,0} \leq S^*_{\beta,\gamma} \leq S^*_{0,\gamma}.$$

In the end the most important problem is how to determine the optimal values S^* and N_{S^*} for fixed λ, μ, γ, β, r and c. If we can prove Conjecture 5.1 and 5.5, we can reduce the number of our necessary calculations a lot. First of all, calculate $S^*_{0,0}$, $S^*_{0,\gamma}$ and $N^0_{S^*}$, which can easily be done by Algorithm 4.1. Now calculate $g^{S,N}_{\beta,\gamma}$ for $S^*_{0,0} \leq S \leq S^*_{0,\gamma}$ and $N^0_{S^*} \leq N \leq N_{\max}$ using Algorithm 5.1. Unfortunately, evaluating $g^{S,N}_{\beta,\gamma}$ using Algorithm 5.1 is not quite fast. As mentioned earlier, more knowledge about the stationairy distribution is desirable, but this goes beyond the extension of this thesis.

6 Conclusion

In Section 3 we described the model with arriving clients, serving agents, a finite queue and abandoning clients from the queue. In Section 4 we calculated the stationary distribution and the long run profit function algebraically. We proved Theorem 4.1 using dynamic programming with value evaluation, which led to the effective Algorithm 4.1 to find the optimal number of available agents and waiting lines for a call centre with certain arrival, abandonment and service rates. The burning question was whether adding a retrial pool, which is a very likely to happen in real life, would change the optimal values for the number of waiting lines and available agents dramatically. In Section 5 we described the new model where abandoning clients were stored in a retrial pool, from which they could re-enter the waiting line at certain rate. Adding the retrial pool drastically changed the structure of the problem in such a way that it was impossible to evaluate the stationary distribution. Therefor we turned to other methods to evaluate the long run profit of the call centre, namely value evaluation using dynamic programming. We formulated the value function and Algorithm 5.1 to evaluate the profit function. We did some experiments wich resulted into several conjectures. Further research on these conjectures goes beyond the extension of this thesis. As we wanted to get more knowledge about solving the problem of optimal planning for inbound call centres, we did get more knowledge. This knowledge led to the insight we are far from the end of research on inbound call centres.

References

[1] L.C.M. Kallenberg. Dictaat besliskunde 3. 2009.

[2] G. Koole. Monotonicity in markov reward and decision chains: Theory and applications. 2007.

[3] G. Koole and A. Pot. A note on profit maximization and monotonicity for inbound call centers. 2009.