

Pacing Algorithm For Outbound Dialing: A Simulation Approach

Nikolay Anisimov*, Konstantin Kishinsky, Nikolay Korolev,
Gregory Pogosyants, Vadim Zyarko

Genesys Telecommunication Labs, Inc.
1155 Market Street, San Francisco
CA 94103, USA
www.genesyslab.com

Abstract – This paper addresses the problem of prediction intervals for outbound calls generated by predictive dialers. More specifically, the problem is to define the time interval for generating the next call based on current and past information so that operators (agents) have enough work and customers are not waiting too long for the next available agent. We will build a model of the system comprising a number of processing servers (agents), a queue, and a dialer defined in terms of queuing system theory. In order to solve the problem we propose the simulation approach. Based on past and current information (e.g., queue length, number of busy agents) the algorithm simulates system behavior and finds the optimal time for generating the next call.

Keywords – predictive dialing, call center, queuing theory

I. INTRODUCTION

Currently, call centers play a very important role in modern business. They perform a wide spectrum of functions such as reservations, sales, customer service, and technical support. A call center can be defined as a system that “consists of a managed group of people working in a computer-automated environment who spend most of their time doing business by phone.” [7]. Usually the people who process calls are called agents. Technically, call centers are based on technology known as Computer Telephony Integration (CTI).

There are two types of call processing in a call center: inbound and outbound. Inbound calls are originated outside the call center by customers, then reach the call center where they are routed to agents. Outbound calls, conversely, originate inside the call center, reach customers and then are processed by agents.

A typical application of outbound calls is a campaign where customers are called by call center equipment.

There are several approaches to outbound dialing where predictive dialing [6] is the most sophisticated and effective one.

This paper addresses the problem of calculating outgoing calls for predictive dialing. The software system that controls the outbound dialing is called Campaign Manager. Roughly speaking, the problem is to define moments of outbound call launching so that agents will have enough work and customers are not waiting long in the queue for an available agent. This task belongs to the area of queuing system theory [3] where there are two main approaches – analytical and simulation [2]. We will consider the *pros* and *cons* of these approaches:

- An analytical approach is very easy to implement. In fact, it is based on using some parameters related to input and output call flow of the system. This approach can be applied in a wide range of the system parameters. It may be especially useful for systems with a large number of servers (agents).
- The main drawback of the analytical approach is that we have to simplify the model of the system to obtain precise formulas for predictive dialing. Fortunately, in many cases it seems that this simplification is not far from a real outbound dialing process.
- On the other hand, simulation modeling reflects real physical processes in a system and therefore can be easily understood by users. That will help in adjusting system parameters and increase confidence in the results of calculation.

*To contact the author: E-Mail: anisimov@genesyslab.com

The simulation approach allows the use of collected information related to call processing in the past. In particular, the information could be used for tuning the parameters of the system (e.g., distribution functions) based on gathered data. This approach makes it possible to take into consideration current information about the system (e.g., queue length, number of ready agents).

- The main disadvantage of the simulation approach (especially when applied to the on-line mode) is the necessity for essential computations that must be performed in a limited time. However, there may be a tradeoff between precision and the speed of computation.

In this paper we consider the simulation approach. The analytical approach reported in [1, 5] is recommended for systems with a large number of agents, because in this case the system behavior is more stationary and may be described with analytical approximations. For a small number of agents we suggest using the simulation approach since it is closer to reality and does not require essential computations.

II. STATEMENT OF THE PROBLEM

The Campaign Manager scheme that will be referred to hereafter as the System is illustrated in Figure 1. It comprises the following elements: a call generator, a dialer, the process that waits for an answer, a customer detector, a queue, and a set of agents. The System operates as follows:

1. The call generator generates call inquiries using a predefined strategy.
2. The dialer is a device that dials customers using the phone number received from the generator.
3. The waiting process waits for an answer from the customer's phone during some pre-defined time (e.g., three dialing tones). If there is no answer during this time interval the waiting process decides that the customer is absent and the call is released. If the customer's phone answers the call is transferred to the Call Progress Detector (CPD).
4. The detector detects the nature of the answer. If it does not represent a live customer (e.g., fax or answering machine) then the call is released. Otherwise the call is delivered to the agents' queue.

5. The agents' queue keeps calls in the order of their arrival, i.e., it implements the procedure "first-in-first-out" (FIFO). If an agent becomes free then he/she receives the next call from the queue. If there are no calls in a queue (the queue is empty) and there are available agents, then the call is directly distributed to an available agent without staying in the queue. A call in a queue may leave the queue before being processed by an agent. This happens when a customer does not wait and hangs up. Sometimes such calls are called nuisance calls [6].
6. An agent processes the call received from the queue. Typically, there are two types of calls: short calls and long calls. Short calls correspond to situations where a customer immediately ends the conversation and hangs up. The typical duration of short calls is one or two seconds. Long calls take place when a customer agrees to take part in conversation with the agent and this typically lasts a few minutes.

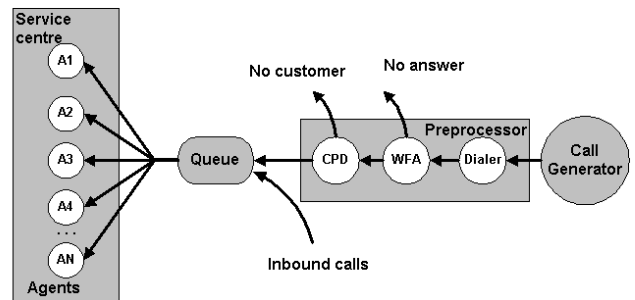


Figure 1: Outbound environment

The three elements Dialer, Waiting For Answer process (WFA) and the Call Progress Detector (CPD) (see Figure 1) cause some delay in call processing. We will refer to these three elements as a call preprocessor. In many call centers agents may also process inbound calls that should be taking into consideration

The problem is to define a strategy for call generation by the call generator in such a way that the following conditions are met:

- All agents must have enough work, i.e., agent utilization should not be less than some pre-specified rate ρ_{min} .
- The rate of customers who have not waited for processing and have left the queue should not be higher than some specified value Ar_{max} . This rate is usually referred to as *an abandon rate*.

It should be noted that in general the task stated above may not have a solution, i.e., the two requirements may not be satisfied simultaneously. Therefore we will solve two separate problems:

- The abandon rate should be maintained at an admissible level Ar_{max} , while maximizing agents' utilization as much as possible.
- Agents' utilization is maintained at an admissible level ρ_{min} while the abandon ratio is as small as possible.

At the moment of calculating the time for the next call the generator has access to the following current information about the System:

- The total number of agents in the System (note that the number may vary). This value will be denoted as m ;
- The number of busy agents;
- The length of the agents' queue, i.e., number of calls in the queue;
- The number of calls in the preprocessor;
- The time that the call is in the preprocessor (for each call).

The generator also has access to some historical information such as:

- Actual time for call processing by agent for each call;
- Actual time for call processing by preprocessor for each call;
- Actual time of staying in the queue of those calls that left the queue without being served by agents.

III. MODEL OF THE SYSTEM

In this section we present a model for the System described above. After that we formulate the criteria of optimization during call generating.

A. Some assumptions

On the one hand the model is a simplified representation of the whole system. On the other hand, the model is a strict formal representation that allows formal treatment and reasoning about its properties and

therefore about the properties of the whole system. The model is described in terms of queuing system theory [3] and its graphical representation is depicted in Figure 2.

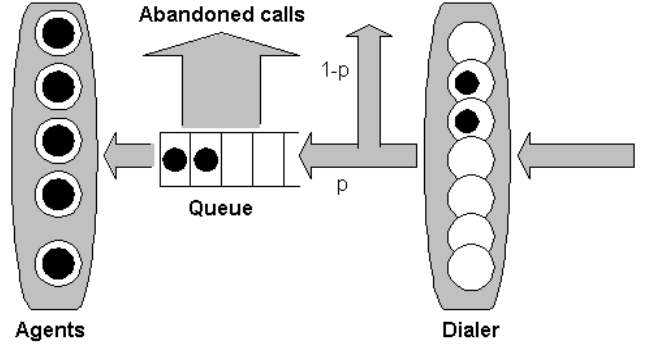


Figure 2: Model

In terms of queuing theory the model is represented as a network of queues that consists of two nodes called server centers that are connected sequentially. One center corresponds to the agents' server and the other one corresponds to the preprocessor.

In this paper we make the following assumptions about the distribution functions of the system:

- Agents' service times have exponential distribution with parameters τ_1 and τ_2 :

$$f_1(t) = \begin{cases} \frac{p_1}{\tau_1} e^{-\frac{t}{\tau_1}} + \frac{p_2}{\tau_2} e^{-\frac{t}{\tau_2}}, & \text{if } t \geq 0; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

- An abandoned process has an exponential distribution with parameter μ ;
- The preprocessor simply delays calls and relates to a deterministic delay process with the delay parameter τ_3 .

Those who are familiar with Kendal's notation of the queuing system [3] will note that the agents' system is a multiserver system with general arrival time and hyperexponential service time: $G/H2/m$. The preprocessor is a multiserver queuing system with loss.

We will use the following System parameters:

- τ_1 – the average time of processing long calls;
- τ_2 – the average time of processing short calls;
- p_1 – the probability of long calls;
- $p_2 = 1 - p_1$ – the probability of short calls;
- M – the capacity of the preprocessor;

- T_p – the delay in the preprocessor;
- p – the probability that a call will reach a living customer and, therefore, will be placed in the queue;
- μ – the average time that customer waits in a queue before abandonment;
- m – the total number of agents;
- ρ_{min} – the minimal agents' utilization rate;
- Ar_{max} – the maximal abandon rate.

We will also use the variables ρ and Ar to represent the agents' utilization and abandon rate, respectively.

Graphically, a circle represents each agent in the agents' center. A service element in the preprocessor (e.g. a line) is also represented by a circle. Tokens placed in circles and queue cells correspond to calls. Therefore in Figure 2 we can see that three agents are busy with three calls, two calls are waiting in the queue, and two calls are in the preprocessor.

B. Criteria of optimization

Now we are prepared to give a precise statement of the problem. Let t_0 be an initial moment of time. At this moment we have the current number of busy agents K_0 , current queue length Q_0 , and number of calls in the preprocessor P_0 . Moreover, for each call in the preprocessor we have the remainder time that the call will stay in the preprocessor: tr_i , $0 \leq i \leq P_0$. We have to calculate the time $t_1 \leq t_0$ when the generator will produce its next call so that two requirements are met:

- the average abandonment rate must not exceed some maximal rate: $\overline{Ar} \leq Ar_{max}$;
- the average agent utilization must not be less than some minimal rate: $\bar{\rho} \geq \rho_{min}$.

It should be noted that in general the task stated above may not have a solution. Indeed, as is known from theory, for all types of queue systems in a stable state the relationship between the average length of queue Q and the utilization of agents ρ looks like the curve shown in Figure 3.

Our requirements $\overline{Ar} \leq Ar_{max}$ and $\bar{\rho} \geq \rho_{min}$ is graphically represented by shaded rectangular. From the graph we can see that it is possible when the curve is not intersected the rectangular.

In this paper we will assume that there are two cases:

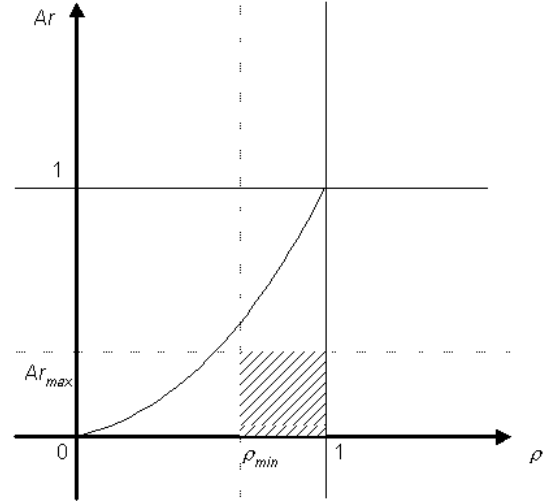


Figure 3: Queue length and agent's utilization

- The abandon rate should be maintained at the admissible level, while maximizing agent's utilization as much as possible.
- Agents' utilization is maintained at the admissible level while the abandon ratio is as small as possible.

IV. PREDICTIVE ALGORITHM

In this section we present a method of calculation for outbound calls. We give the necessary mathematical background and the algorithm derived from it.

A. Mathematical background

First, let us consider processes in an agents' service center. We will estimate two random processes of queue length and number of busy agents. The first is closely related to abandon rate and the last is related to agent utilization.

The process for defining the average length of a queue has the following form:

$$Q_t = Q_0 + \int_0^t I\{K_s = m\} dN_s - D_t - S_t, \quad (2)$$

where

1. Q_0 is a queue length at the initial moment of time;
2. Control process N_s makes the next jump at the moment of entering the next call from a preprocessor;

3. The process $D_t = \int_0^t \mu Q_s ds$ represents the average abandonment of a client from the queue.
4. The process $S_t = \int_0^t m I\{Q_s > 0\}(\frac{p_1}{\tau_1} + \frac{p_2}{\tau_2}) ds$ describes the average processing of the calls.
Here $I\{x\}$ is an indication function with $I\{x\} = 1$ when x is *true* and $I\{x\} = 0$ if x is *false*.

Less formally, the formula (2) can be explained as follows. Q_t denotes an average number of calls in the queue at time t , i.e. Q_t is a function of time. The first member of the right side of the equation is the initial number of calls at the beginning ($t = 0$). The second member describes the number of calls which left the queue to be distributed to agents within time interval $[0, t]$. Roughly speaking, we sum all incoming calls that are generated by the process N_s but only when all agents are busy (otherwise a new incoming call directly proceeds to an available agent without changing the queue). The third member calculates all calls that left the queue due to an abandon process (customer hangs up). The number of abandoned calls depends on the queue length Q_s and parameter μ characterizing the intensity of the call leaving. The last member calculates calls that left the queue due to transferring to an agent. To accomplish this, we sum all calls that have been processed by agents provided that the queue at the time of finishing the call is not empty.

The process describing the average number of busy agents is as follows:

$$K_t = K_0 + \int_0^t I\{Q_s = 0, 0 \leq K_s \leq m\} dN_s - \int_0^t I\{Q_s = 0, 0 < K_s \leq m\} K_s (\frac{p_1}{\tau_1} + \frac{p_2}{\tau_2}) ds. \quad (3)$$

The members of the equation have the following meaning:

1. K_0 is the number of busy agents at the initial moment of time;
2. The second term in the equation corresponds to average number of the calls arriving from the pre-processor and being delivered directly to an available agent (i.e., the queue is empty);
3. The third term corresponds to the average number of freeing agents after finishing call processing (long or short) provided that the queue is empty (and therefore an agent will not be captured by the next call).

Less formally, the formula (3) can be explained as follows. It calculates the number of busy agents at time t . To do so we calculate the arrived calls from the dialer when the queue is empty and there are ready agents (second member). We also subtract all calls that have been completed by agents when the queue is empty (otherwise the agent immediately receives a new call from the queue).

Let us consider the behavior of the system in the absence of incoming calls. The equations (2) and (3) are transforming into the following:

$$Q_t = Q_0 - D_t - S_t, \quad (4)$$

and

$$K_t = K_0 - \int_0^t I\{Q_s = 0, 0 \leq K_s \leq m\} K_s (\frac{p_1}{\tau_1} + \frac{p_2}{\tau_2}) ds. \quad (5)$$

Let us derive the formula for abandon rate Ar . Abandon rate is defined as a relation of the abandoned call to all calls that left the queue during some time interval. Clearly $0 \leq Ar \leq 1$. Let $Q_1 - Q_0 = \Delta Q$ be the variation of the queue during the time interval $\Delta T = t_1 - t_0$. Let also ΔQ_D be the number of calls that left the queue due to abandon during ΔT . Then the average abandon rate for the interval ΔT can be calculated as follows:

$$\overline{Ar} = \frac{\Delta Q_D}{\Delta Q} \quad (6)$$

To calculate the average agent's utilization $\bar{\rho}$ at the interval ΔT we will use the formula:

$$\bar{\rho} = \frac{\sum_{i=1}^n K_i}{n \cdot m}, \quad (7)$$

where K_i is the average number of busy agents at the interval $\Delta t_i = t_{i+1} - t_i$ and n is the number of intervals Δt_i within the interval ΔT .

B. Method of calculation

B.1 General principles

The main idea of our approach is as follows. Starting from time point t_0 we simulate the processes Q_t and K_t in the absence of calls from the generator. Clearly, after a time these processes will decrease and tend towards zero. At each step of the simulation we estimate the abandon rate Ar . Based on this estimation we determine the optimal time for generating a new call by the generator.

In simulation of these processes we will use the so-called *fixed-incremented time advance* approach [2] for advancing the simulation clock. According to this approach, the simulation clock is advanced in moments of exactly Δt time units for some appropriate choice of Δt . After each update of the simulation clock we calculate increments of K_t and Q_t and update them. We also calculate related parameters like abandon rate and agent utilization. The value of Δt is chosen to be small enough to make the probability of multiple events (e.g. arrivals of calls) be negligible, yet large enough that processing time is not excessive. Therefore there should be a reasonable trade-off based on practical experience.

The fixed-incremented time advance approach seems to be more preferable for our case with respect to the *next-event time advance* approach because we need on-the-fly simulation to estimate the time interval comparable with the next event time.

B.2 Generating random events

In simulation we will need to generate events having some distribution function. We will assume that we can generate random variations of $U[0, 1]$ within the interval $[0, 1]$. Let $F(t) = P\{T \leq t\}$ be the continuous distribution function of random variable T . Then the conditional probability of the occurrence of an event within the time interval $[t, t + dt]$ is

$$P\{T \in [t, t + dt] \mid T \geq t\} = \frac{dF(t)}{1 - F(t)}. \quad (8)$$

Then in order to generate an event in interval $[t, t + \Delta]$ for small enough Δ we calculate interval $\alpha = \frac{\Delta F}{1 - F(t)}$. Clearly, $\alpha \leq 1$. Then generate random value $x = U[0, 1]$ and check whether $x \leq \alpha$. If yes, then we generate event. Otherwise, not.

It is not hard to see that for the case of exponential distribution, i.e. $F(t) = 1 - e^{-\frac{t}{\tau}}$, we have

$$P\{T \in [t, t + dt] \mid T \geq t\} = \frac{1}{\tau} dt \quad (9)$$

That is in the case of exponential distribution the probability does not depend of the time.

B.3 Algorithm for simulation Q_t and K_t

The algorithm of simulation processes Q_t and K_t works as follows. At initial moment it receives current value of queue length Q_0 and number of busy agents K_0 . The processes are calculated by sequence of steps, each step being corresponded to time increment Δt .

At each step processes Q_t and K_t are calculated as a meaning of N realizations of discrete processes $Q^i(t)$ and $K^i(t)$, $i \in \{1, \dots, N\}$. Each realization of $Q^i(t)$ and $K^i(t)$ is calculated according to equations (4) and (5). The algorithm stops when processes reach the state where target criteria is satisfied.

B.4 Method for simplified case

To better understand the proposed method, let us start with a simplified case. Consider a model derived from the general model (see Figure 2) by removing the preprocessor center. The resulting model is depicted in Figure 4.

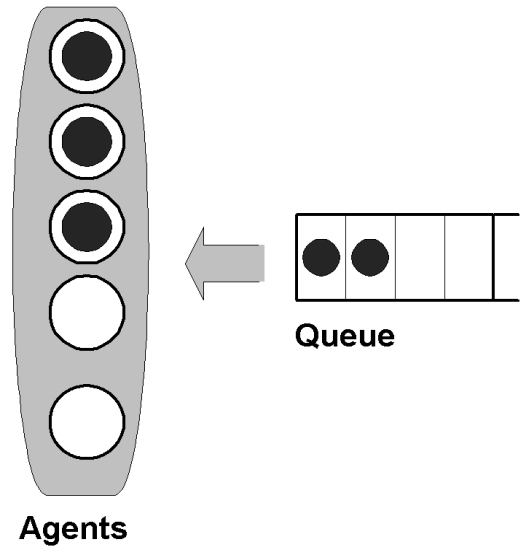


Figure 4: Simplified model

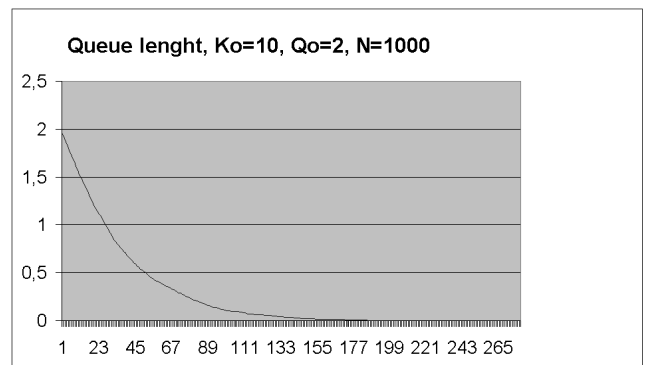


Figure 5: Queue length, $K_0 = 10, Q_0 = 2, N = 1000$

We will observe the system behavior in the absence of incoming calls starting from the time t_0 . To accomplish this we simulate the processes Q_t and K_t starting from current values Q_0 and K_0 according to

formulas (4) and (5), respectively. In particular, after each Δt we calculate increments ΔQ and ΔK , new values of Q_t and K_t . Clearly they will be decreased.

On Figures 5 and 6 the processes Q_t and K_t are depicted. In this experiment we supply the system with the following parameters: $m = 10$, $p_1 = 0.7$, $p_2 = 0.3$, $\tau_1 = 120$, $\tau_2 = 5$, $\mu = 0.1$. The initial values are $K_0 = 10$, $Q_0 = 2$. The number of realization is $N = 1000$.

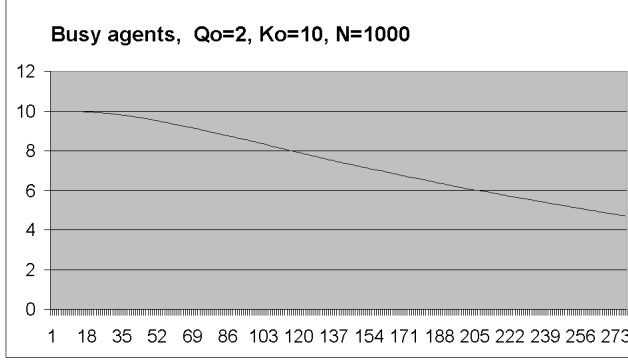


Figure 6: Busy agents, $K_0 = 10$, $Q_0 = 2$, $N = 1000$

We will try to find the “optimal” moment for the generation of the next call using processes Q_t and K_t . It is clear that in stationary case there will be an optimal queue length Q_{opt} . Suppose we reach time t_1 where the length of the queue is equal to optimal Q_{opt} . The next call should be generated at time t_2 , which is induced by one call¹, as in the following:

$$Q(t_1) - Q(t_2) + m - K(t_2) = 1 \quad (10)$$

Here we use a fluid approximation considering that the part of the call $m - K(t_2)$ directly transfers to agents and the rest $1 - (m + K(t_2))$ is placed in the queue². The optimum of the interval $[t_1, t_2]$ means that during the interval the abandon rate is equal to the maximal value Ar_{max} :

$$Ar(t_1, t_2) = \frac{\Delta Q_D}{\Delta Q + m - K(t_2)} \leq Ar_{max} \quad (11)$$

Here we also suppose that part of the new call $m - K(t_2)$ is transferred to agents and therefore improves the abandon rate.

¹We can also use a fluid approximation and set call length equal to hit ratio p bearing in mind that this part of the call passes through the dialer and $1 - p$ part of the call is cancelled

²Clearly, in this case $m - K(t_2) \leq 1$.

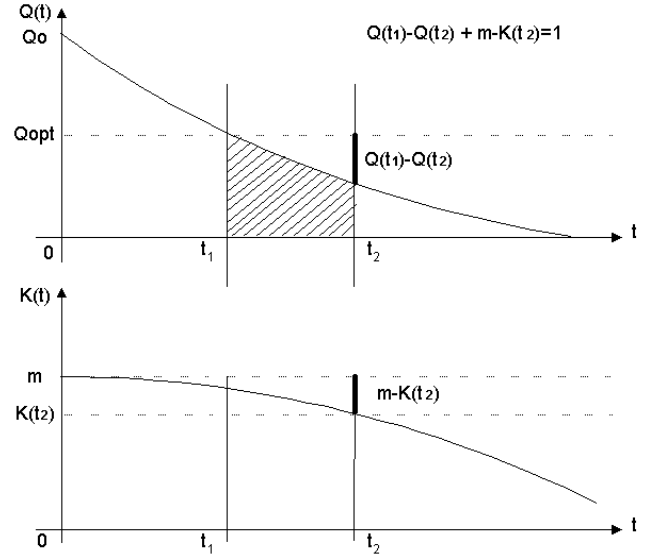


Figure 7: Finding the optimal interval. $Q_0 \geq Q_{opt}$.

Therefore to find the optimal time t_2 for generating the next call we move the interval $[t_1, t_2]$ from 0 to the right until conditions (10) and (11) are satisfied. Note that the length of the interval $[t_1, t_2]$ will vary. The time t_2 will be the time for the next call.

All of the above assumes that the initial queue length is greater than optimal, i.e., $Q_0 \geq Q_{opt}$. Suppose that $Q_0 \leq Q_{opt}$. Let us continue the curve of the queue to the “past”, see Figure 8, and find the optimal interval $[t_1, t_2]$. There may be two cases.

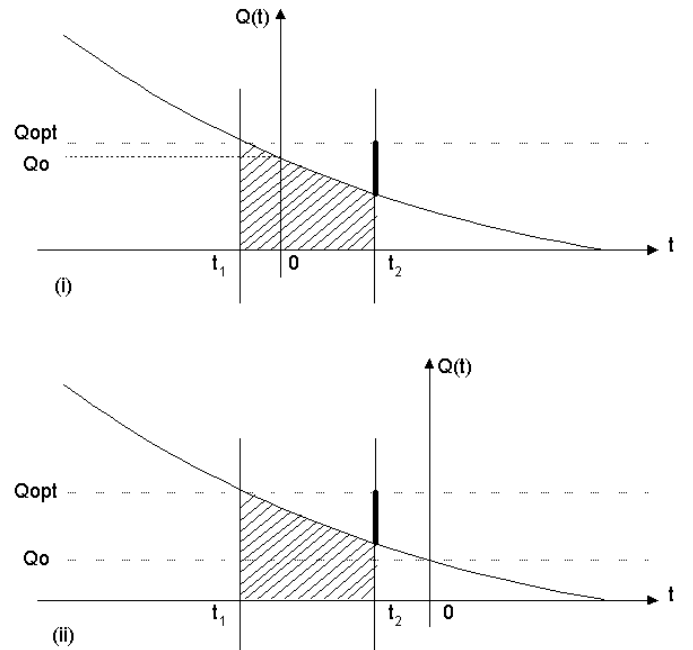


Figure 8: Finding optimal interval. $Q_0 \leq Q_{opt}$.

Case 1: $t_1 \leq 0, t_2 \geq 0$ (Figure 8(i)). In this case the point t_2 is the optimal point.

Case 2: $t_1, t_2 \leq 0$ (Figure 8(ii)). In this case optimal time t_2 is in the past. But we can not generate calls in the past and therefore all we can do is to generate the call immediately³.

Now, taking into consideration our inability to generate more than one call, we suggest the following procedure.

1. Simulate processes Q_t and K_t starting from the initial values Q_0+1 and K_0 , respectively.
2. When we reach some t' such that $Q(t') = Q_0$, remember t' .
3. Find the optimal interval $[t_1, t_2]$ as defined above.
4. If it is impossible, i.e., if $Ar(t_1, t_2) > Ar_{max}$ for all t_1 and t_2 , take $t_{opt} = t_0$. If we find $[t_1, t_2]$ such that $t_2 \geq t'$ then $t_{opt} = t_2 - t'$. If $t_2 \leq t'$ then $t_{opt} = t_0$.

If we want to optimize (i.e. maximize) agents' utilization to some boundary value ρ_{min} while the abandon rate is as low as possible, we can also employ the technique described above. However, instead of using the formula (11), we calculate agents' utilization within the interval $[t_1, t_2]$ as follows:

$$\rho(t_1, t_2) = \frac{\sum_{i=0}^M K(t^i)}{M} \geq \rho_{min} \quad (12)$$

where the interval $[t_1, t_2]$ has M intervals Δt with time points $t^1, t^2, \dots, t^{(M+1)}$, $t_1 = t^1$, $t_2 = t^{(M+1)}$.

B.5 Method for general case

Let us consider the general case. The model includes the preprocessor that may contain calls at the initial point of time. These calls should be taken into consideration during simulation.

Let us modify the above methods as follows:

1. During simulation of processes Q_t and K_t we will count arriving calls from the preprocessor. More specifically, each call in the preprocessor is equipped with the time of entry into the preprocessor. At each time t_i we check whether the call should leave the preprocessor or not.

³If we have possibility to generate more then one call at a time, we can generate $Q_{opt} - Q_0 = Q(t_2) - Q(t_0) \geq 1$ calls. But this is contradicts our assumption about ability to generate no more then one call at a time.

2. The transfer of a call from the preprocessor to the queue can be modeled by a generator of random variantes without any problems. Alternatively, we can exploit fluid approximation and change the queue $Q'(t_i) = Q(t_i) + (p - (m - K(t_i)))$ and the number of busy agents $K'(t_i) = m$ if $m - K(t_i) \geq p$. If $m - K(t_i) \leq p$ then $K'(t_i) = K(t_i) + p$ and $Q(t_i)$ is not changed.
3. When the last call from the preprocessor is transferred to to queue (or cancelled) at the time t_f the simulation continues similarly as in simplified case. At this stage we obtain the optimal time t_{opt} .
4. Finally, calculate the resulting time as follows: $t'_{opt} = t_f + t_{opt} - T_p$ if $t_f + t_{opt} \geq T_p$. Otherwise we take $t'_{opt} = t_0$ and generate the next call immediately.

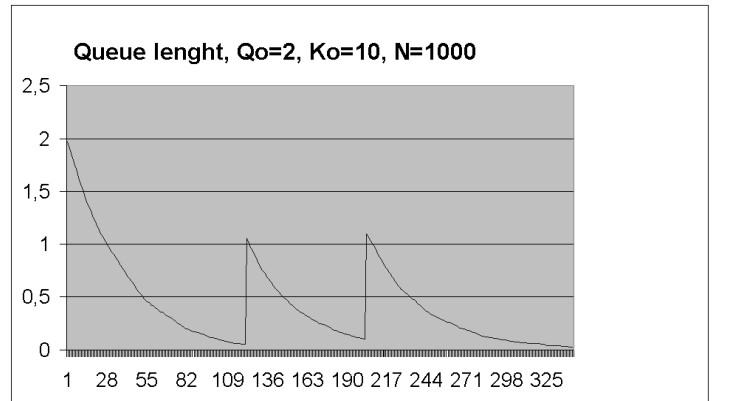


Figure 9: Queue length, $K_0=10, Q_0=2, N=1000$

In Figure 9 we show the process Q_t in the system with three calls in the preprocessor. Only two calls reached the queue while the other one has been canceled (e.g., no answer) in the preprocessor.

V. CONCLUSION

In this paper we presented simulation approaches to outbound call generation to be used in predictive dialers. We focused on a simplified case where the call processing and call waiting time have exponential distribution and the preprocessor possesses a fixed delay.

We can point out some direction for further developing this approach:

1. Moving from idealistic distribution functions to more realistic distribution functions. There may be two possibilities:
 - Selecting the most appropriate distribution function from those widely used in practice (e.g. normal, Erlang, Gamma, Weibull) and fitting the parameters of the distribution.
 - Making use of empirical distribution built on previous experience.
2. Developing methods of adaptive control over system parameters and distribution functions, in particular. For instance that would be of great importance when we do not know the precise total number of agents (e.g., agents are also involved in other activities such as processing inbound calls).

REFERENCES

- [1] N.Anisimov, N.Korolev, G.Pogosyants, Y.Shtivelman, V.Zyarko. Methods for Generating Outbound Calls in A Call Center Environment. Submitted to GLOBECOM'2000.
- [2] A.M.Law, W.D.Kelton. Simulation Modeling and Analysis, McGraw-Hill, Inc., 1991.
- [3] L. Kleinrock. Queuing Systems. Volume I. Theory. Wiley, New York, 1972.
- [4] H. Kramer. Mathematical Methods of Statistics, Princeton, Princeton university press, 1946.
- [5] Predictive Algorithms for Generating Outbound Calls. Internal Technical Report. Genesys Telecommunication Labs., Inc. November 1999.
- [6] A. Szlam, K. Thatcher. Predictive Dialing Fundamentals. Flaitron Publishing, Inc. New York, 1996. ISBN 0-936648-80-5.
- [7] R.Walters. CTI in Action, John Willey & Sons. 1997. ISBN 0 471.