

Predictive Dialer Intensity Optimization Using Genetic Algorithms

Pedro M. T. Amaral and Miguel M. Vital

Abstract—Companies rely on contact centers to act as communication links with their clients. Outbound dialing is often used to reach existing or new customers. This task is generally performed by automatic dialers, which initiate new calls depending on the amount of working agents. The probability of a customer answering a call, however, depends on a set of conditions, such as the time schedule or the type of day. This fact presents itself as a challenge to automatic dialers, since contact lists with low answer probability can make the contact center's agent occupation rate very low. Predictive dialers tackle this problem in an automated way by generating more calls than the number of available agents. The majority of predictive dialer algorithms use statistical approaches to adjust the automatic dialer intensity, which is used to decide on the amount of calls that should be initiated at each time. In this paper, we propose a method of optimizing the automatic dialer intensity using genetic algorithms – evolutionary methods based on natural selection and genetics. We implement the proposed algorithm by modifying the current proprietary Altitude Software predictive dialer and perform a comparative evaluation between both versions. Our method obtained superior results to those achieved by the original algorithm, with a slightly higher agent utilization rate.

Index Terms—Contact centers, dialer intensity optimization, genetic algorithms, predictive dialers.

I. INTRODUCTION

Contact centers, the natural successors of the old telephony call centers, play an increasingly important role on today's business world. Millions of agents across the globe work on such facilities, which serve as a customer-facing channel for firms in many different industries [1]. For many companies, such as airlines, hotels, and retail banks, contact centers serve as the primary link between the customer and the service provider [2].

There are two main types of interactions in a contact center: (a) inbound interactions, initiated by customers outside the contact center; and (b) outbound interactions, originated manually or automatically inside the contact center, with the purpose of reaching new or existing customers. Both these interaction types are handled by agents, who act on behalf of the company that owns or contracts the contact center services [3].

Most academic research related to contact centers focuses on pure inbound environments, where only inbound

interactions are being handled. Previous studies address topics such as management strategies to deal with impatient customers [4], and agent schedule optimization in order to improve the contact center service quality [5], [6].

In contrast with these previous publications, this paper focuses solely on outbound interactions, namely those that are automatically initiated by the contact center. Automatic outbound dialing can be performed using a wide set of dialing methods [7], from which we highlight the following three: preview, progressive (or power), and predictive dialing.

Preview dialing can be categorized as a semi-automatic dialing method, since the contact to be executed is presented beforehand to the agent, who is able to decide whether or not the call should be made. Though this method is able to achieve the best interaction experience (by allowing the agents to prepare conversations or decline contacts that may not result in acceptable business outcomes), it has a very low agent utilization rate, especially in situations where the probability of a call being answered by a customer is low.

Progressive dialing initiates a new call for each agent that finishes an interaction. For each non-answered call, a new one is initiated. This cycle repeats until all agents are busy handling interactions. In progressive mode, the amount of active calls (either the ones being handled or the ones waiting for a customer) never exceeds the number of working agents. The agent utilization rate is better than the one achieved with the preview method, but it still suffers from the negative impact of a low answer probability.

Predictive dialers were designed to overcome the unsatisfactory agent utilization rate problem present in both preview and progressive dialing. In a predictive mode, the key idea is to anticipate when an agent will finish its interaction and to proactively initiate calls, so that a new customer will answer shortly after the agent becomes idle [8]. Therefore, predictive dialers allow more active calls than the number of agents that are ready to handle them. This can lead to what is commonly designated as nuisance calls – calls that are disconnected by the dialer shortly after the customer answers, because no agent is available to handle the interaction. Some countries impose strict limitations on the amount of nuisance calls that can be made by contact centers within a certain time frame [8]. The predictive dialer pacing mode is further described in Section II.

Some sophisticated strategies involve combining these methods throughout the life of the outbound campaign. It is common, for instance, to use a progressive approach during the start-up phase of the campaign, leaving predictive dialing for a posterior stage. This enables the collection of some initial operational statistics that can be used to prepare for a more advanced dialing mode [9].

Manuscript received September 20, 2013; revised December 10, 2013. This work was supported in part by QREN (Quadro de Referência Estratégico Nacional).

Pedro M. T. Amaral and Miguel M. Vital are with Altitude Software, Oeiras, Lisbon, Portugal (email: Pedro.Amaral@altitude.com, Miguel.Vital@altitude.com).

Predictive dialers usually make decisions mainly based on statistical information. Some widely tested models that have already been proven successful in the past, such as the use of queuing theory and Poisson processes [8], [9], while the application of other decision making methodologies has been left unexplored. One such methodology, machine learning is nowadays widely used to solve complex problems involving the extraction of relevant and non-trivial information from data.

In this paper, we propose a new method that applies machine learning, through genetic algorithms, to the problem of optimizing the dialing intensity of a predictive dialer module. Section II covers some state of the art regarding predictive dialers. Section III describes genetic algorithms, their main phases, and the properties that make them suitable to our problem. Section IV explains how we adapted the current predictive dialer running at Altitude Software to use genetic algorithms. Section V details the configuration used in the experiments that were performed on the proposed method. Finally, Section VI presents the results obtained, and Section VII concludes with some future work suggestions.

II. PREDICTIVE DIALERS

Samuelson [8], who claims to have invented predictive dialing in the late 1980s, described these dialers as methods that determine when computer-directed outbound telephone dialing systems should initiate a call. Generally, the amount of generated calls is based on (i) a prediction of how many agents will be free after the calls are initiated by the dialer and (ii) the estimated success probability of these calls being answered by a customer [9].

The performance of a predictive dialer can be measured by how closely it meets the following principles:

- Maximize agent utilization time;
- Keep the nuisance rate – the amount of nuisance calls divided by the amount of calls answered by customers – below a certain threshold.

In order to succeed, predictive dialers need to have access to a set of contact center data, such as the probability of a customer answering a call, the history of call durations, and the amount of working agents.

Predictive dialers must also be robust enough to properly handle sudden changes in the contact center's state, such as variations in the number of ready agents due to shift changes or fluctuations in the call answer probability throughout the different time schedules of the day. Adaptation to such events should be nearly instantaneous.

Despite the low amount of literature devoted to automatic outbound dialing [9], we are able to distinguish two main predictive dialer method approaches: analytical based and simulation based.

Analytical based predictive dialing generally relies on queuing theory [10]. Call volume forecasts are translated into λ values for Poisson arrivals, and average interaction durations into μ values for exponential service times [11]. Korolev *et al.* [9] proposed a predictive dialing method where a $M/G/m/m$ queuing system is used to model the outbound calls generation rate. The author assumes that the outbound

call flow can be modeled as a Poisson process with parameter λ and that the interaction duration (T_s) has a general distribution with parameter $\mu=1/T_s$. One of his approaches consists in estimating the maximum dialer intensity that can to keep the nuisance rate close to the requested (N_R). To accomplish that, the author uses the Erlang- B formula [12], as shown in (1), where α is the number of working agents, and $\rho = \lambda \times T_s$ is the traffic offer – number of calls that are initiated in each interval of time.

$$E_B(\alpha, \rho) \leq N_R \Leftrightarrow \frac{\rho^\alpha}{\sum_{i=0}^{\alpha} \frac{\rho^i}{i!}} \leq N_R \quad (1)$$

This method was shown to perform well in managing the nuisance rate when the number of agents was above 50, decreasing in performance as that number was lowered to below 20. Additionally, it does not take into consideration the agents' busy status.

Simulation based predictive dialing is a technique that uses real contact center operational data to run simulations and estimate the best performing automatic dialing rate. Filho *et al.* [13] studied the impact of using simulated future market behavior to estimate the predictive dialer rate. According to the authors, the dynamic and non-deterministic circumstances in which predictive dialers operate, with a high number of stochastic variables to deal with, make analytical models perform below expectations. They developed a real-time simulation model that, given some contact center historical data regarding a certain period t_{n-1} , estimates the best dialing rate for period t_{n+1} . To simulate the future market behavior, the authors used a chi-square fitting method on the stochastic variables that were considered most relevant for the problem. Since it was not the core of the study, the simulation strategy and methods were not described. The results obtained showed that the approach based on future market behavior simulation was able to achieve results similar to the ones obtained by a commercial predictive dialer, in relation to the following metrics: agent utilization time and observed nuisance rate.

Our proposed solution follows a simulation based strategy using genetic algorithms to perform the estimation of the dialer intensity of a predictive dialer module. This class of algorithms is described in the following section.

III. GENETIC ALGORITHMS

Genetic algorithms, first described by John Holland in the 1960s [14], are search procedures inspired by natural selection and genetics [15]. Included in the field of evolutionary computation, genetic algorithms have a wide set of applications, which range from protein folding simulations [16] to optimizing investment strategies [17].

A genetic algorithm is a method for evaluating a population of entities called chromosomes, which consist of a set of genes – candidate values for a certain problem variables. A traditional genetic algorithm is divided into three steps: selection, crossover, and mutation. The selection step is responsible for sorting the population according to a fitness

function, which uses a set of criterion to infer the quality of a chromosome. Crossover, as the name suggests, combines two chromosomes, whose probability of being crossed is usually proportional to their fitness values. Finally, during mutation, the chromosomes that resulted from the crossover suffer random modifications to their genes. This process is repeated for a certain number of iterations called generations. Further details regarding genetic algorithms can be found in Mitchell's article [18].

Genetic algorithms are very noise tolerant, not prone to overfitting, and do not use a large amount of problem-specific information [15], [19]. Moreover, they are highly parallelizable – either at the level of the generation or when computing the fitness criteria – and adaptive, in the sense that they keep performing well in a changing environment [18]. All these factors make the application of genetic algorithms in a predictive dialing environment very attractive.

IV. METHOD DESCRIPTION

The purpose of this research is to evaluate how genetic algorithms perform as dialing intensity optimizers in a predictive dialer environment. For this purpose, we used the predictive dialer algorithm in production at Altitude Software with some minor modifications, as shown in Fig. 1.

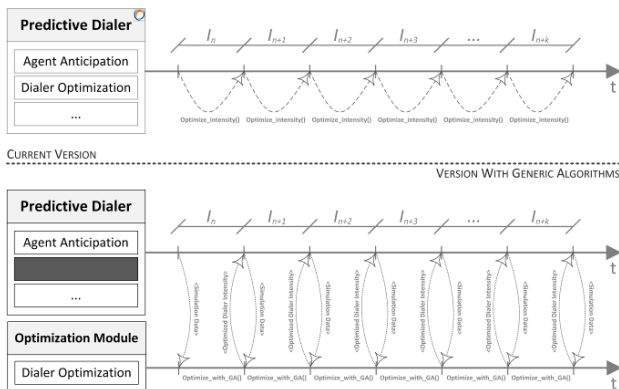


Fig. 1. Dialer intensity optimization using genetic algorithms.

The Altitude Software predictive dialer is aware of a set of contact center operational data, such as the amount of dialed calls, the answer probability, the number of ready agents at each time. With such information, this module is able to compute the amount of calls that need to be initiated, and the adjustments that need to be applied to the dialer intensity (I_j). The current strategy used to optimize the latter is out of the scope of this report.

The method that we propose differs from the original algorithm at the level of the dialer intensity optimization. Instead of computing the adjustments to this property inside the predictive dialer module, the new version of the algorithm sends some simulation data to a new optimization module, which is responsible for computing the new dialer intensity. The result of this operation is then retrieved to the predictive dialer module to be used in the future call estimation operations. The optimization module uses an implementation of a genetic algorithm, which is further detailed ahead.

The simulation data that is sent to the optimization module contains all the information that is needed to replay the

contact center behavior of a previous time interval. If we consider that time is divided into fixed-length intervals (T_i), we can state that we use simulated data corresponding to instant T_{i-1} to optimize the dialer intensity of instant T_{i+1} , since the simulations occur during instant T_i . Simulation data should allow us to replicate events such as: agent state changes, call durations, script path nodes that were executed, wrap-up durations, intervals of time between call starts and customer answers, and so on. Contrarily to Filho *et al.* [13], we do not estimate future contact center behavior. We assume that past behavior can be used to adapt to the future and that if we use small time interval lengths we can still react to sudden market changes sufficiently fast. Notwithstanding, we leave the evaluation of the impact of such an enhancement to future investigations.

The remaining of this section concerns the specifications of our genetic algorithm. The nomenclature used was extracted from [15], [18].

Our chromosome representation is different from usual. Instead of using gene chains consisting of arrays of bits, our chromosomes have just one real-valued gene, corresponding to the dialer intensity under evaluation.

The fitness function evaluates the performance that the contact center would achieve if it used each of the chromosomes in the population. This requires us to replay the previous contact center behavior, using the simulation data, for each chromosome. The fitness of a chromosome is then a combination of the nuisance and agent utilization rates that it achieved during the simulations.

To cross two chromosomes, we use a strategy based on weighted means. We compute a random weight, ω , (between 0 and 1) and use it to generate the offspring genes (G):

- $G_{\text{Offspring}\#1} = (\omega \times G_{\text{Parent}\#1}) + ((1 - \omega) \times G_{\text{Parent}\#2})$
- $G_{\text{Offspring}\#2} = ((1 - \omega) \times G_{\text{Parent}\#1}) + (\omega \times G_{\text{Parent}\#2})$

For the mutation, depending on the mutation probability, we stress the offspring genes by adding a random value between -50% and 50% of the current gene value.

Genetic algorithms are typically iterated for anywhere from 50 to 500 or more generations [18]. However, since we are dealing with real time optimization and since we perform simulations using previous contact center behavior, we use just 20 generations. We compensate this low amount of generations with more frequent genetic algorithm executions. This way, we can benefit from the optimization properties of this type of algorithms, while providing quicker intensity updates to the predictive dialer module.

Each generation runs with 11 chromosomes. The initial population is obtained using the previous best chromosome as reference, being composed of: the previous best chromosome, five chromosomes with lower gene values, and the remaining with higher values.

V. EXPERIMENTAL SETUP

To test our approach, we performed a set of trials using the following configuration. Each experiment lasted 5 hours, with a requested nuisance ratio below 5%. Only one pure outbound campaign was used in each test. The initial state of the

campaign was reset in the beginning of each simulation, meaning that prior to that moment no interaction had ever been made in the campaign's context. In order to mimic a more realistic environment, we forced the probability of a call being answered by a customer to vary in the following fashion: in the first 90 minutes it was 20%, in the second it was 50%, and in the last 2 hours it was 30%. The agents responsible for handling the calls were computer controlled. The interaction flow depended on a campaign script that generated both short and long calls (representing, for instance, failed or miss-targeted sells vs. successful sells). Short calls lasted between 26 and 38 seconds, while long calls lasted from 37 to 73 seconds. Optimizations to the dialer intensity ran in periods of 20 minutes. Moreover, the campaigns never went short on the amount of contacts ready to be initiated. No skill profiles or contact segmentation were used in our analysis.

This configuration was used in three distinct scenarios:

- *20_*: Using a constant amount of 20 agents.
- *50_*: Using a constant number of 50 agents.
- *MIXED_*: Using a different amount of agents in each hour of the test, namely: 10, 20, 50, 30, and 15 (that is, 10 agents in the first hour, 20 in the second, and so on).

The first two scenarios were designed to test the impact of the new approach using a small (*20_*) and a decent amount of agents (*50_*) – predictive dialers tend to perform worse when the amount of agents is below 20, and start becoming efficient when this number is above 50 [9]. The last scenario is intended to simulate the entrance and exit of agents in the contact center, since the amount of working agents is one of the properties that most affects the optimization of the dialer intensity. Together with variations in the answer probability, these three scenarios allow to simulate an environment that is very close to the reality in most contact centers.

Since we applied genetic algorithm optimization to the current Altitude Software predictive dialer module, we analyzed the effectiveness of our approach regarding the performance of the original method. For this purpose, we ran each solution under the previous three scenarios: *20_ALT*, *50_ALT*, and *MIXED_ALT* (using the current implementation of the Altitude Software predictive dialer) and *20_GEN*, *50_GEN*, *MIXED_GEN* (using the method proposed in this report).

In the next Section we compare the results obtained, regarding the following metrics:

- Obtained nuisance rate, calculated as shown in (2), where *#Nuisance* is the amount of observed nuisance calls and *#Handled* is the amount of calls that were answered by clients and handled to agents.

$$N_{Observed} = \frac{\#Nuisance}{\#Nuisance + \#Handled} \quad (2)$$

- Average agent idle time, as computed in (3), where *idle_time_Agent_i* is the total idle time of agent *i*, and *#Agents* is the number of agents.

$$Avg(Agent IdleTime) = \frac{\sum_{i=1}^{\#Agents} idle_time_Agent_i}{\#Agents} \quad (3)$$

Although the purpose is to keep the observed nuisance rate very close to the requested value, that may not always occur, since the predictive dialer may not need to be excessively aggressive in order to keep the agents busy.

Maximizing agent average utilization time is equivalent to minimizing the agent average idle time. Predictive dialers must try to keep this property as high as possible, depending on the requested nuisance. If the agent utilization time is below the expectations and if the observed nuisance is smaller than the requested, the automatic dialer should try to follow a more aggressive approach.

Notwithstanding, it is important to note that average idle time can be very low in the early life of a campaign, since there is not much margin for mistakes. During this period, a single nuisance call can significantly increase the observed nuisance rate to larger values than the ones that are allowed by some countries' law.

VI. RESULTS AND DISCUSSION

Fig. 2 shows the results obtained in our experiments. We present the performance of both methods in each scenario – *20_*, *50_*, and *MIXED_* – regarding the observed nuisance ratio (on the left) and the average agent idle time (on the right). The blue solid lines correspond to our solution, while the orange dotted lines correspond to the current Altitude Software's predictive dialer. In the average idle time graphs, we also present a zoomed in version – smaller graph with grey background on the upper right corner – with the performance observed in the last hour of the each experiment.

In the *20_* scenario, where we used a constant number of 20 agents, our solution has made a larger amount of nuisance calls, thus achieving a higher nuisance ratio than the other method. Neither algorithm exceeded the maximum allowed nuisance ratio. Regarding agent utilization, both approaches were able to keep the interval of time that an agent had to wait for a customer to answer below 18 seconds. However, the method with genetic algorithms achieved lower idle times. In this scenario, our solution has slightly superior performance than the current Altitude Software's method.

These results are similar to the ones achieved using the scenario with 50 agents. Our approach made more nuisance calls than Altitude Software's predictive dialer, but achieved lower average idle times (a beneficial tradeoff in most business scenarios). The mean inactivity times were, once again, less than 18 seconds in the last hour, with the method using genetic algorithms showing average times lower than 10 seconds in the last half hour.

In the mixed scenario results, we can see how the algorithms adapted to the variations in the number of working agents. When, in the end of the first hour, the number of agents doubled (from 10 to 20), our solution made a larger number of nuisance calls, but was also able to achieve higher agent occupation times. The new algorithm observed a lower nuisance ratio in the beginning of the second hour (when the number of agents was 50), but kept increasing the agents' occupation. This result is a consequence of the increasing number of agents, which led to more calls being handled by an

agent before the contact center had to disconnect them. Both the nuisance ratio and agent occupation decreased in the last two hours, when the number of agents decreased to 30 and 15, respectively. The original Altitude Software’s predictive dialer showed an opposite behavior. It verified less nuisance calls when the number of agents was higher, and a larger amount when that number decreased. Regarding agent occupation, although the results were very similar, Altitude Software’s method performed better than our method with fewer than 50 agents. Based on these results, we can assert that our approach performs better when the number of

working agents increases, and that its predecessor works better when this number decreases.

In the beginning of all experiments, we can observe high average idle times with any call resulting in nuisance. This occurs since no historical operational data is available when the campaign starts. During this initial period, the predictive dialer must act similarly to a progressive dialer, by never initiating more calls than the number of available agents. The occurrence of a nuisance call during the start-up time has a huge impact in the nuisance call ratio, since the number of handled calls is still very low (see Fig. 2)

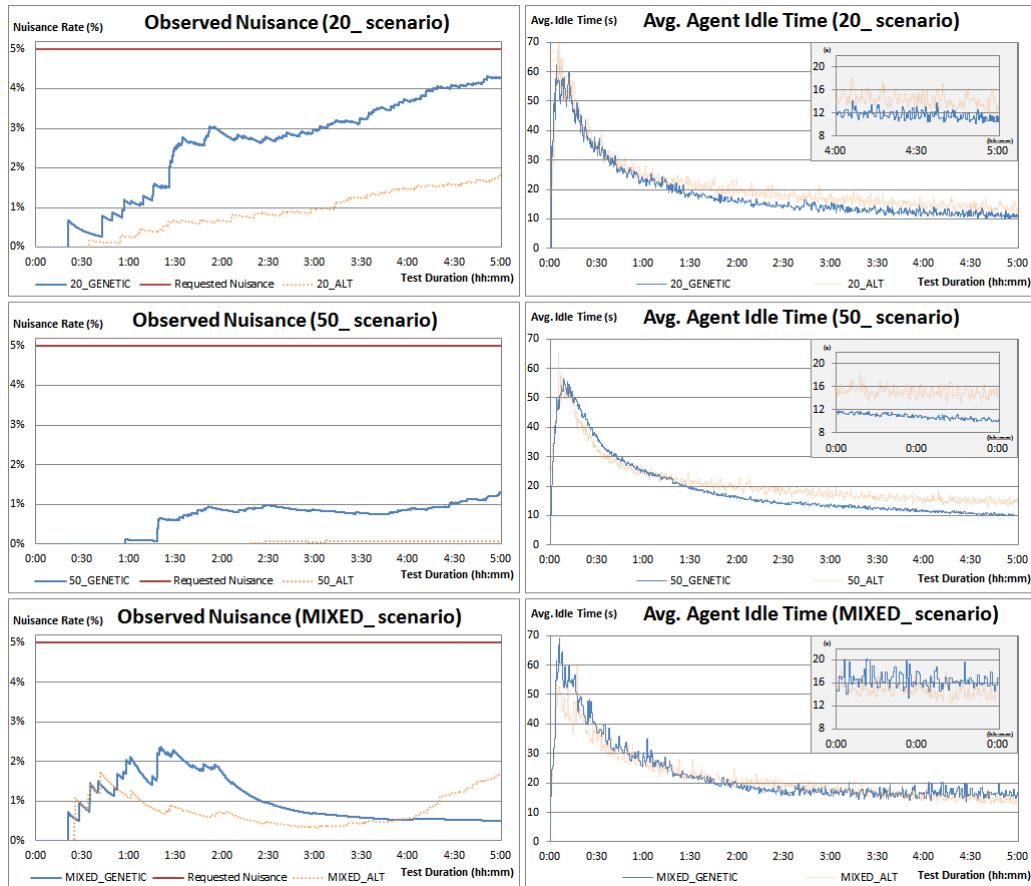


Fig. 2. Evaluation results.

Variations in the call answer probability also produced some impact in the performance of both algorithms, mainly when it increased from 20% to 50% after the first hour and a half. In any case, we can see that both algorithms reacted well, by controlling the growing amount of nuisance calls. The decrease in the answer probability did not produce any significant impact in the performance of either method.

Overall, our solution achieved a slightly superior performance, when compared to the current Altitude Software predictive dialer. We had expected, however, that the agents’ occupation in the scenario with 50 agents would be much higher than when we used just 20. We are certain that this result can still be achieved if our method adopts a more aggressive strategy, since the observed amount of nuisance calls at that time was very low and 50 agents can quickly generate large amounts of handled calls. In any case, if we look to the results that Altitude Software’s current predictive dialer obtained in the same scenario, we can see that it performed worse than our solution, while maintaining a

nuisance call ratio very close to 0%. This may suggest that the small differences in the agent utilization time between these two scenarios may be related to some constraints the predictive dialer may impose on the number of calls that are initiated at each time, independently of the dialer intensity in use. In any case, we are satisfied with the significant improvement of approximately 2 seconds that our proposed solution achieved.

VII. CONCLUSION AND FUTURE WORK

Contact centers play an important role in today’s business world, by acting as a crucial communication channel between companies and clients. Either as a sales method or for quality control, automatic outbound dialing presents itself as an effective solution to reach large sets of customers in short timeframes. Variations to the probability of a call being answered by a customer, either due to different time schedules

or day types, impact the performance of an automatic dialer. Contact lists with small answer probability generate low agent utilization rates, which is undesirable in most business scenarios. Predictive dialers try to overcome this problem by anticipating the amount of agents that are about to finish their interactions and initiating an amount of new calls that will allow them to get a new customer as soon as they finish dealing with the current. Predictive dialers base their computations on a variable called dialer intensity, which is typically optimized using statistical methods.

In this paper, we propose a method based on genetic algorithms to optimize the dialer intensity of the current Altitude Software's predictive dialer. Included in the evolutionary computation category, genetic algorithms use natural selection, crossover, and mutation techniques to perform tasks such as searches and optimizations. The genes in our chromosomes represent the candidate values that should replace the current dialer intensity. Through simulations using historical contact center operational data, our solution computes the effectiveness of each of these genes. In the end, it retrieves the value that obtained the best results during the simulation to be used in future calls estimation.

Our experiments show that our solution is able to achieve superior results to those obtained by the current Altitude Software's predictive dialer, regarding observed nuisance ratio and agent utilization rate. Independently of the environment in which the tests were executed, our method did not produce a nuisance ratio larger than the requested value (5%). Moreover, the average time that an agent had to wait for a new customer to answer was between 10 and 20 seconds after the initial campaign's start-up time. These results can be further improved, especially in situations in which there exists a large number of working agents (≥ 50).

In the future, we plan on using other machine learning algorithms to optimize the predictive dialer performance. These include applying other methods to the optimization of the dialer intensity and to the predictive dialer calls estimator. Additionally, we intend on trying an approach similar to the one followed in [13], and analyze if estimating future market behavior makes a positive impact on our modified predictive dialing method. Predictive dialers must evolve according to the market needs and governmental regulations, which will, in the future, surely bring forth new challenges for these automatic call generation systems. We hope our contribution inspires further work in this subject.

ACKNOWLEDGMENT

We would like to thank Adriano Alves Simões for his contribution in the design of the genetic algorithm.

REFERENCES

- [1] Z. Aksin, A. Mor, and V. Mehrotra, "The modern call-center: a multi-disciplinary perspective on operations management research," *Production and Operations Management*, vol. 16, no. 6, pp. 665-688, November-December 2007.
- [2] N. Gans, G. Koole, and A. Mandelbaum, "Telephone call centers: tutorial, review, and research prospects," *Manufacturing & Service Operations Management*, vol. 5, no. 2, pp. 79-141, March 2003.
- [3] *Contact Center Overview*, Altitude Software, Lisbon, 2009, pp. 3-4.
- [4] A. Brandt, and M. Brandt, "On the M(n)/M(n)/s queue with impatient calls," *Performance Evaluation*, vol. 35, pp. 1-18, March 1999.

- [5] R. Stollatz, *Performance Analysis and Optimization of Inbound Call Centers (Lecture Notes in Economics and Mathematical Systems)*, Springer Verlag, 2003.
- [6] A. Fukunaga, E. Hamilton, J. Fama, D. Andre, O. Matan, and I. Nourbakhsh, "Staff scheduling for inbound call and customer contact centers," *AI Magazine*, vol. 23, no. 4, pp. 30-40, 2002.
- [7] A. Szlam and K. Thatcher, *Predictive Dialing Fundamentals: An Overview of Predictive Dialing Technologies, Their Applications, and Usage Today*, CRC Press, 1996.
- [8] D. A. Samuelson, "Predictive dialing for outbound telephone call centers," *Interfaces*, vol. 29, no. 5, pp. 66-81, October 1999.
- [9] N. Korolev, H. Ristock, and N. Anisimov, "Modeling and simulation of a pacing engine for proactive campaigns in contact center environment," in *Proc. 2008 Spring Simulation Multiconference*, San Diego, 2008, pp. 249-255.
- [10] R. B. Cooper, *Introduction to Queueing Theory*, 2nd ed. New York: North Holland (Elsevier), 1981, ch. 5.
- [11] V. Mehrotra and J. Fama, "Call center simulation modeling: methods, challenges, and opportunities," in *Proc. 2003 Winter Simulation Conference*, San Francisco, 2003, vol. 1, pp. 135-143.
- [12] L. Kleinrock, *Queueing Systems, Volume 1: theory*, Wiley-Interscience, 1975, ch. 6.
- [13] P. J. F. Filho, G. F. Cruz, R. Seara, and G. Steinmann, "Using simulation to predict market behavior for outbound call centers," in *Proc. 39th Simulation Conference*, Washington, 2007, pp. 2247-2251.
- [14] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, 5th ed. MIT Press Cambridge, 1998.
- [15] D. E. Goldberg, "Genetic and evolutionary algorithms come of age," *Communications of the ACM*, vol. 37, no. 3, pp. 113-119, March 1994.
- [16] R. Unger and J. Moul, "Genetic algorithms for protein folding simulations," *Journal of Molecular Biology*, vol. 231, no. 1, pp. 75-81, May 1993.
- [17] R. J. Bauer, *Genetic Algorithms and Investment Strategies*. 1st ed. Wiley, New York, 1994, ch. 3.
- [18] M. Mitchell, "Genetic algorithms: An overview," *Complexity*, vol. 1, no. 1, pp. 31-39, 1995.
- [19] D. E. Goldberg, K. Deb, and J. H. Clark, "Genetic Algorithms, Noise, and the Sizing of Populations," *Complex Systems*, vol. 6, pp. 333-362, 1991.



Pedro M. T. Amaral was born in Lisbon, Portugal. He received his BSc. and MSc. degrees in Information Systems and Computer engineering from Instituto Superior Técnico (IST), Lisbon, Portugal, in 2007 and 2012, respectively.

He is currently working at Altitude Software, Lisbon, Portugal, in the predictive dialer department. Prior to joining Altitude Software, as part of his MSc. thesis, he collaborated at KDBIO, INESC-ID, Lisbon, in a project that aimed to improve prognostics on patients with Lateral Amyotrophic Sclerosis using data mining techniques. This collaboration resulted in a publication at HI-KDD, entitled "Predicting the Need for Non-Invasive Ventilation in Patients with Amyotrophic Lateral Sclerosis". His research interests include machine learning, data mining, and evolutionary computation.



Miguel M. Vital was born in Lisbon, Portugal. He received his BSc. degree in Electronics and Telecommunication from Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal, in 1993. He received his BSc. and MSc. degree in Computer Science from Faculdade de Ciências e Tecnologia, Lisbon, Portugal, in 1995 and 2001, respectively. He completed the PAGE – Advanced Management

Program for Executives – from Universidade Católica Portuguesa, Lisbon, Portugal in 2001.

He is currently with the VP Research and Development at Altitude Software, Lisbon, Portugal. Prior to that, he was R&D Director at Altitude Software, founding partner of Holos S.A., and researcher at UNINOVA. His research interests include software engineering and data mining. Eng. Vital is the member no. 3233 of the Portuguese Engineers Association.