A Novel Methodology for Optimizing Display Advertising Campaigns Using Genetic Algorithms

Article *in* Electronic Commerce Research and Applications · November 2017

DOI: 10.1016/j.elerap.2017.11.004

Luis Miralles
University College Dublin

Hiram Ponce
Universidad Panamericana

Lourdes Martinez-Villaseñor
Universidad Panamericana Sede México

A Novel Methodology for Online Advertising Campaigns Optimization

Luis Miralles-Pechuán[a], Hiram Ponce[b,*], Lourdes Martínez-Villaseñor[b]

[a] *Centre for Applied Data Analytics Research (CeADAR), University College Dublin, Belfield, Dublin 4, Ireland*
[b] *Universidad Panamericana, Facultad de Ingeniería, Augusto Rodin 498, Ciudad de México, 03920, México*

Abstract

Online advertising campaigns have attracted the attention of many advertisers willing to promote their business on the Internet. One of the main problems faced by advertisers, especially by those who have little experience in Internet advertising, is configuring their campaigns in an efficient way. To configure a campaign properly it is required to select the appropriate target, so it is guaranteed a high acceptance of users to adverts. It is also required that the number of visits that satisfy the configuration requirements is high enough to cover the advertisers' campaigns. Thus, this paper aims to present a novel methodology for online advertising campaigns optimization by using genetic algorithms as the main optimization method, and a machine learning model of the click-through rate (CTR) as part of the fitness function. Results showed that the proposed methodology is feasible to optimize online advertising campaigns by selecting the set of the best features required. Also, customization of the advertising campaign selecting some features by an advertiser, e.g. applying micro-targeting, can be optimized efficiently.

Keywords: online advertising campaigns, optimization, genetic algorithms, micro-targeting, machine learning

*Corresponding author
Email addresses: luis.miralles@ucd.ie (Luis Miralles-Pechuán), hponce@up.edu.mx (Hiram Ponce), lmartine@up.edu.mx (Lourdes Martínez-Villaseñor)

1. Introduction

Internet advertising campaigns have experienced a tremendous increase in recent years. The growth in the number of users and in the number of hours they spend connected has caused that the business volume of this sector increases year-over-year (Goldfarb, 2014). In addition, online campaigns offer advertisers interesting advantages that traditional campaigns can hardly do.

For example, online campaigns can be constantly optimized. Advertisers have a real-time feedback of their campaigns which can be reconfigured at any time. This allows advertisers to reduce or to increase the campaign budget according to its performance. Another important advantage of these platforms is that they offer advertisers the possibility to configure many parameters such as age, time, browser, operating system or device type, among others, which allows advertisers to address their campaigns to a very specific public. This technique is known as microtrageting and, due to its high performance, it is widely applied in advertising campaigns (Goldfarb and Tucker, 2011). Additionally, online campaigns can be afforded by any budget: from small businesses that spend a few dollars a day to large multinationals that make marketing expenses of millions of dollars.

Ad networks are responsible for managing advertisers' campaigns, earning a commission for that service. Ad networks main tasks are: distributing the adverts between publishers' websites (Miralles-Pechuán et al., 2017), and preventing fraud in the online advertising ecosystem (Wilbur and Zhu, 2009). To this end, it is necessary to develop tools to efficiently manage all the processes involved in advertising campaigns. These tools should provide a friendly interface, keeping in mind that they have to be used by publishers and advertisers with limited technological knowledge.

One of the problems faced by small ad networks is that they have serious difficulties in offering advertisers profitable campaigns, since it is very difficult to launch targeted campaigns at a much selected group when ad networks have small volume of visits (Provost et al., 2009). Therefore, campaigns are generically addressed and, consequently, have lower performance than the targeted ones. For this reason, many small ad networks have disappeared and many others are at risk of disappearing (Evans, 2008).

In that sense, this paper aims to present a novel methodology for online advertising campaigns optimization, mainly focused on small ad networks. In order to configure a campaign efficiently, we identified that it is required to display the most interesting adverts for users and that the number of visits that meet the configuration requirements is sufficient to cover the advertisers demand. Moreover in our proposal, the settings of online campaigns are optimized according to two basic constraints: (i) the interest of users for the adverts, and (ii) the number of visits that matches those settings. In this regard, the interest of users can be measured as the average click-through rate (CTR), i.e. the probability that a user generates a click on an advertisement (Richardson et al., 2007), of all predicted visits. Thus, it is assumed that when a user performs clicks, it is because the advert raises in users some interest. In addition, we are not looking for configurations that include a huge number of visits, since advertisers campaigns are usually not massive in small ad networks. But we are seeking configurations that ensure a sufficient number of visits to meet the demand of the majority of the advertisers.

To reach that goal, we propose a hybrid soft computing method based on genetic algorithms (GA) and supervised machine learning (ML). This method can be adapted to any ad network properly modeling ad campaigns, and setting suitable parameters in the optimization process.

Based on that, the proposed methodology uses GA as the optimization method, to generate individuals that represent campaign configurations. Each individual is evaluated at each iteration of the algorithm, based on a heuristic fitness function, taking into account the estimated number of visits matching the campaign configuration and the estimated average CTR of those visits. Actually, it is required to develop a CTR prediction model with the best predictive power as possible, to be used in the estimation of the average CTR at the fitness function. To develop such a model, we used the online logistic regression method as the supervised ML since its performance has been widely demonstrated (McMahan et al., 2013; Chapelle et al., 2015).

Our novel methodology allows small ad networks to suggest advertisers profitable configurations for their campaigns. It is also easy to implement and cost-effective in terms of space-time complexity. Thus, the contribution of this work considers: (i) the development of a general methodology for online advertising campaigns optimization, (ii) the implementation of machine learning for CTR modeling as part of the objective function in the optimization problem, and (iii) a method that provides mechanisms to fit the requirements of custom ad campaigns, i.e. micro-targeting, such as the relevant features and the expected number of visits.

The rest of the paper is organized as follows. In Section 2, we describe the state of the art of online advertising campaigns. In Section 3, we present the proposed methodology for campaigns optimization using genetic algorithms and an estimation machine learning model. Section 4 describes two experiments that were conducted to set the parameters of the proposal, and Section 5 presents the results and discussion of the work. Lastly, Section 6 concludes the paper and suggests future work.

2. Online Advertising Campaigns

In this section, a review of the online advertising campaigns and online advertising optimization is described.

2.1. Ecosystem for Online Advertising

Digital advertisement, including online and mobile advertising, has experienced a sharp growth year-over-year, according to Interactive Advertising Bureau (IAB) (IAB, 2016). In the ecosystem for online advertising, publishers have ad spaces in their webpages and advertisers spend money to place their ads on those spaces. Large advertisers can directly negotiate with large publishers. Nevertheless, direct collaboration is commonly done through brokers including ad networks and ad exchanges. Ad networks provide intermediation services between publishers and advertisers. Ad exchange enterprises provide auctions for ad spaces similar to a traditional stock exchange (Chen et al., 2016). The main issue is how to solve the continuous massive matching problem between users and advertisers in order to obtain the best performance and return-of-investment (ROI) for the whole ecosystem. On the Internet, a large number of advertisers deliver multiple messages to an enormous number of consumers (Evans, 2009). There is a variety of strategies and technologies to solve the matching problem. The two more used types of advertising are search-based advertising and display-advertising. In search advertising, advertisers and consumers are matched based on keywords entered in a search query.

The most relevant ads are displayed as sponsored links resulting from a query. The most commonly pricing models in search advertising are cost per click (CPC), cost per action (CPA), and cost per lead (CPL). In display advertising, different types of ads are displayed on publisher websites. Recently, real-time bidding is used to display the most relevant ad impression for the consumer based usually in demographic information. Display advertising is mainly related with cost per mille (CPM) pricing model, but it also includes CPC, CPA, and CPL (Chen et al., 2016). Contextual and behavioral advertising are also related to display advertising. Contextual strategies take into account the browser query to identify opportunities to show ads related to the current search. One challenge is how to match the content of a site with the product or service advertised. Big digital advertising stakeholders play several roles of digital advertising monopolizing the online advertising market. The largest Internet advertisement companies have great advantages when compared with small ad networks (Evans, 2008):

- *Volume and market* – when a product is released to the market it can be oriented to win money by margin or volume. Large ad networks have understood that Internet business is not in the margin but in volume. Few people are willing to pay for an Internet service, but millions are willing to use it for free. Millions of users imply opportunities to show them ads and make money. On the other hand, it is very difficult for a business to survive with few users.

- *Synergies* – when a large company develops a module, this module can be incorporated into many of its products. Thus, users can be directed to websites through ad networks' search engine; but also, these large networks can manage the adverts of the websites.

- *Financing* – large companies are so powerful and have such high incomes from different sources, therefore they are less likely to have funding problems. So, they can launch platforms focused on increasing users but making no economic benefits, and might capitalize the users volume.

- *Resistance to change* – once a user is accustomed to using a platform, or has used a specific email service, for that user to change, he has to overcome an adaptation barrier.

The main advantage of a large ad network is that it has almost the entire market, and it has a captive audience of billions of users. The aforementioned advantages hamper the development of small and new ad networks. Online advertisement campaigns have gradually been oriented towards more specific niches. Small advertising networks have been disappearing given their inability to offer advertisers campaigns with high performance for a specific target due to their scares number of visitors.

2.2. Online Advertising Optimization

Every stakeholder of the ecosystem for online advertising wants to make a profit from ad exchanges. Research to optimize revenue from each point of view has been done: buyers (advertisers), sellers (publishers), and intermediaries (ad networks). In this paper, we focus in the buyers' point of view considering the relation between advertisers and ad networks. Advertisers spend a budget placing advertising messages; they have to determine how to spend their money in order to maximize the effectiveness of the advertising campaign. Hence, research has been done to optimize budget distribution in online advertising (Aronowich et al., 2014). Muthukrishnan et al. (2007) formulate stochastic version of budget optimization problem. Perlich et al. (2012) combine machine learning techniques as well as a second price auction theory to determine "the correct price to ensure that the right message is delivered to the right person, at the right time". Lee et al. (2013) focus in online bid optimization. They presented an online approach to optimize the performance metrics while satisfying the smooth delivery constraint for each campaign. Other authors focus on optimizing advertiser satisfaction, on the understanding that advertisers will be more willing to make investments if they get good profit. Balseiro et al. (2014) make an in-depth analysis on the balance that must exist between economic performance, the most profitable ad selection, and the quality that is offered to advertisers.

In recent years, advertisers have become increasingly demanding with the requirements to reach a specific target. Advertisers can segment their campaigns using various attributes such as user demographic characteristics, city or geographical area, session time, keywords, device, operating system, browser, etcetera. This is known as microtargeting, which reduces the number of visits that may meet the requirements of advertisers, but in turn makes these visits have a greater value and relevance (Provost et al., 2009; Sivadas et al., 1998). With the proper segmentation of users, advertisers can place their ads to specific groups increasing the probability of interest in their services or products. Large companies that frequently have several roles in online advertising, have captive consumers and obtain a lot of information of user's profile and behavior from several platforms. This is valuable information for consumer segmentation and microtargeting purposes. Small ad networks cannot offer such targeted campaigns because they do not receive enough visit information; only a small segment of consumers actually comply with the requirements of advertisers. Hence, small ad networks campaigns are usually general oriented and low performance.

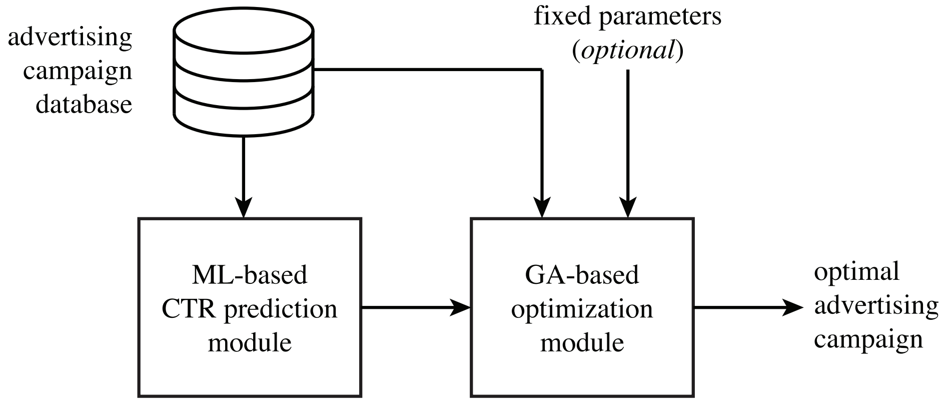

Figure 1: Schematic of the proposed methodology for online advertising campaigns optimization.

3. Description of the Online Advertising Campaigns Optimization

In this section, we describe our proposal. The methodology for online advertising campaigns optimization consists on a hybrid soft computing method based on genetic algorithms and a supervised machine learning model, aiming to determine the best subset of features that maximizes the CTR of a given advertising campaign.

In that sense, the methodology can be adapted to any properly ad campaings models and to set suitable parameters in the optimization process. Figure 1 shows the proposed methodology. The overall optimization procedure is computed by genetic algorithms. Then, individuals of the GA correspond to campaign configurations, and at each iteration these individuals are evaluated using a fitness function. Particularly, this methodology proposes a heuristic fitness function that optimizes both the number of visits and the average CTR of a campaign. To this end, the online logistic regression model (McMahan et al., 2013; Chapelle et al., 2015) is used as a supervised machine learning to model the estimated average CTR of a campaign. Details of the proposed methodology are described below.

3.1. Design of the Prediction CTR Model

Consider a dataset of online advertising campaigns with D samples (rows) and N numerical and/or categorical features, f_i, (columns). Each sample describes one user's exposure to an advert, represented by a set of features related to the ad, the user and the metadata. For example, these features can be: the operating system, browser, age, time, type of product, among others. An additional column, y, in the dataset represents the real value that a user clicks (1) or not (0) into the online advert. In that sense, the CTR model aims to predict the probability CTR value, p, of a visit, such that, $p = P(y|f_i)$.

There are many researches focused on accurately predicting the CTR of an advert (Shan et al., 2016; Lee et al., 2012). Our goal is to implement a model as precise as possible in predicting the CTR. At the same time, it is highly recommended to build a very cost-effective model in terms of time and memory computing resources. This methodology can be easily implemented on small ad networks at a low cost and in a short period of time. For this reason, we have decided to implement the CTR estimation model proposed in (McMahan et al., 2013) that is based on the online logistic regression method. The model is explained in detail in many scientific papers, e.g. (Chapelle et al., 2015; Li et al., 2011). Currently, this model applies the hashing trick technique (Li et al., 2011) and the adaptive learning rate (Chin et al., 2015).

In a nutshell, the hashing trick is an ingenious method to model data sets with large quantities of information using a hash function. The latter transforms the categorical and numerical features of each entry into an integer value within a range between zero and D. It is highly recommended to use a large number for D in order to avoid collisions. Collisions happen when different original values generate the same number after applying the hash function. Hash values are used as the array index. The benefit of applying the hashing trick is that it reduces the spatial dimensions, and therefore, the memory and the time required to create the model. The hashing trick method has become famous for its simplicity and its effectiveness (Li et al., 2011).

The algorithm for building the online logistic regression model uses two arrays, n and w; where n is an array of integers that represents the number of times each feature appears after applying the hashing trick and w is an array of real values that represent the weights associated to each feature. The values for w are updated using (1); where, $w[i]$ represents the weight of the i-th feature (initially set to zero), $n[i]$ represents the number of times the i-th feature appears after applying the hashing trick, $y \in \{0, 1\}$ is the target output value, $p \in [0, 1]$ is the estimated output value, and α represents the heuristic adaptive learning rate for optimizing the online logistic regression model.

$$w[i] = w[i] - \frac{\alpha(p-y)}{\sqrt{n[i]+1}} \tag{1}$$

Once w and n, the weight and frequency arrays, are totally updated, the output for each entry, p, is predicted using the sigmoid function of (2).

$$p = \frac{1}{1 + exp\left(-\sum_{i=1}^{N} w[f_i]\right)} \tag{2}$$

3.2. Design of the Genetic Algorithms for Online Advertising Campaigns

Genetic algorithms is a metaheuristic optimization procedure that implements simple operations observed in the adaptation and evolution of species. The general strategy of genetic algorithms considers to generate a population of individuals, e.g. a set of possible solutions, that evolves through generations (iterations of the algorithm).

At each epoch or generation, the set of individuals are evaluated aiming to determine which of them are the best candidate solutions by employing a fitness function. The latter is used to measure the adaptability of each individual in order to select the most appropiate ones. Once that, crossover and mutation operations are computed over the selected individuals with given probabilities, i.e. probability of crossover p_c and probability of mutation p_m. Then, at the end of each epoch, the current population is updated with the new individuals created from the selection, crossover and mutation operators. Algorithm 1 shows the general strategy of genetic algorithms. To this end, individuals are coded in order to perform the operators described above.

Algorithm 1 Simple genetic algorithm.

```
initialize population
while termination criterion is not reached do
    evaluate population using fitness function
    select individuals
    perform crossover and mutation
    update population
end while
```

In that sense, we propose the design of the genetic algorithms for online advertising campaigns considering the following: individual encoding, fitness function definition, and determination of the selection, crossover and mutation operators.

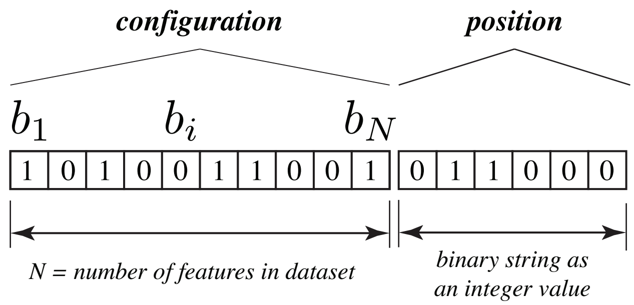

Figure 2: Individual encoding scheme represented with two chromosomes: configuration and position.

3.2.1. Individual Encoding

We propose an individual to represent an advertising campaign configuration that consists on a subset of the original dataset.

Consider a dataset of D samples and N features. Then, an individual is encoding with two chromosomes, as depicted in Figure 2. The first chromosome is a binary string, $\{b_1 \dots b_i \dots b_N\}$, that represents the selected features in a particular configuration, where bit "1" in position i indicates that the i-th feature is selected and bit "0" that it is not. For example, if a dataset is composed of $N = 4$ features, then the chromosome value "0101" determines that features f_2 and f_4 should be selected.

The second chromosome represents the fixed values that a particular advertising configuration should have, and it is encoded as a binary representation of one integer value, q, in the range $[0, D-1]$. In fact, this integer represents the q-th sample in the original dataset. For example, if the dataset has $D = 5$ samples, then the chromosome value 2 ("010") refers that the third sample in the dataset should be selected.

To this end, the combination of the chromosomes, configuration and position, represents the online advertising campaign configuration by selecting the subset of samples that has the same values as the q-th sample only in the selected features such that $b_i = 1$ for all $i = 1, \dots, N$. This procedure is exemplified in Figure 3.

3.2.2. Fitness Function

The GA seeks for configurations that maximizes two variables: the average CTR and the number of visits in an online advertising campaign. Thus, we propose a fitness function that evaluates an advertising campaign heuris-

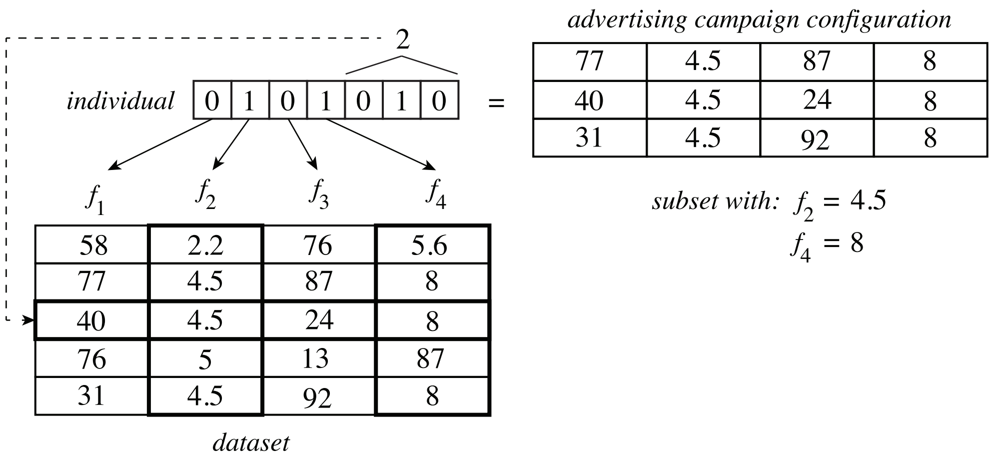

Figure 3: Example of an online advertising campaign configuration using the individual.

tically by considering: the number of visits to the campaign calculated as the number of samples in the subset, D', and the estimated average click-through rate, $CTR_{average}$, as written in (3); where, T is a threshold value to avoid solutions with larger number of visits.

$$f(individual) = CTR_{average} \times \min(D', T) \tag{3}$$

Particularly, the average CTR is computed by evaluating each sample k of the subset into the machine learning model (2) to estimate the CTR value p_k and obtaining the mean value of these predictions, as expressed in (4).

It should be noticed that we are looking for an enough number of visits to satisfy the expectations of the advertisers, taking into account that most advertisers have a limited budget. Thus, we are trying to obtain configurations which generate a high average CTR value, and with a number of visits that simply equals or exceeds, albeit slightly, the threshold value T. In this work, we propose that T should be set manually by the ad network or the advertiser.

$$CTR_{average} = \frac{1}{D'} \sum_{k=1}^{D'} p_k \tag{4}$$

3.2.3. Selection, Crossover and Mutation Operators

Once the individuals are evaluated using the fitness function, a selection mechanism is computed to determine the best adapted individuals that will pass to the next generation. In that sense, we propose to use the linear-rank selection method (Scrucca, 2013) since it assigns a better probability to be chosen to those individuals that are ranking first. This selection method prevents premature convergence of the algorithm. In addition, an elitism strategy is also considered so that the best E individuals of one generation are chosen to be part of the next generation. This approach ensures that the best invidual so far is considered to be part of the population at any generation (Scrucca, 2013).

Then, we propose to use the single-point crossover operator (Scrucca, 2013) that chooses two selected individuals with probability p_c and randomly selects a crossover point so that the portions of the two strings beyond this point are exchanged to from two new strings.

Lastly, each new individual is subjected to a uniform random mutation operator (Scrucca, 2013) that selects each gene of the individual (i.e. a bit string) and changes the bit with probability p_m with its complement binary value.

4. Experimentation

This section describes the experimentation process in order to validate our proposal. First, a brief description of the dataset used is presented. Then, we explain the process to build and train the CTR model. Lastly, we describe the methodology of the experiments conducted in this work to optimize online advertising campaigns.

4.1. Description of the Dataset

In this paper, we use a dataset of ad campaigns for mobile phones (Avazu, 2015). It corresponds to 10 days of click-through data organized in more than 40 million samples, and it is composed of a set of columns that represents features from the users and the websites. Twenty-four attributes are comprised in the dataset: ad identifier, click ("0" for non-click and "1" for click), hour, banner position, site identifier, site domain, site category, application identifier, application domain, application category, device identifier, device IP address, device model, device type, device connection type, and other nine anonymized categorical variables.

Table 1: Performance metrics of the prediction CTR model over the testing set.

metric	**value**
accuracy	0.8352
logarithmic loss	0.3967
root-mean squared error (RMSE)	0.3524

4.2. Implementation of the Prediction CTR Model

Firstly, we built and trained the prediction CTR model, as described in Section 3.1. Thus, we randomly selected 12-million samples from the original dataset: the first 10-million samples for training and the remaining 2-million samples to evaluate the prediction CTR model. In addition, we converted all features from strings to integer values, and we applied module 2^{20} to calculate the hashing table. Lastly, we applied the online logistic regression method to build the prediction CTR model with the following settings, chosen manually: learning rate $\alpha = 0.1$, and $D = 2^{20}$ as the length of arrays n and w. Algorithm 2 summarizes the implementation of this CTR model for both training and testing it. To carry out the CTR model, we use R Studio environment 3.3.2.

Different performance metrics were calculated to the resultant CTR model, as depicted in Table 1. In this case, the CTR-model response computes 39.67% in the logarithmic loss metric, just 1.76% worse than the best prediction CTR model obtained with the 40-million original dataset (Avazu, 2015); representing a well prediction CTR model that can be used for further experimentation.

4.3. Configuration of the Genetic Algorithms

The next step was to configure the setting parameters of the genetic algorithms. Particularly, an exploration of the probability of crossover p_c and mutation p_m parameters was conducted. Due to large computational time, we employed only 20-thousand samples randomly chosen from the testing dataset for the preparation of the setting parameters.

To do that, we set 9 different values for p_c and p_m in the interval $[0.1, 0.9]$ with steps of 0.1. Then, we performed a set of 30 experiments for each combination of p_c and p_m values, and the average of the best fitness evaluations of that experiments were reported, as shown in Table 2. Additionally, the population size was set to 500 and the percentage of elistism was 5%. For the threshold value in the fitness function, we selected $T = 20$. Notice that

Algorithm 2 Training and testing the prediction CTR model.

Require: dataset
Ensure: prediction CTR model

$data \leftarrow$ Randomly select 12M samples visits from the original dataset
$data_{hash} \leftarrow$ Apply the hashing trick to $data$
Divide $data_{hash}$ into *training* and *testing* sets ▷ 10M *training* and 2M for *testing*
$D \leftarrow 2^{20}$ ▷ length of w and n
$\alpha \leftarrow 0.1$
$w \leftarrow \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$ of length D
$n \leftarrow \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$ of length D

▷ Training the CTR model
for all $v_k \in training$ **do** ▷ v_k is a *training* sample
    $s \leftarrow 0$
    for all $f_i \in v_k$ **do**
        $s \leftarrow s + w[f_i + 1]$
    end for
    $p_k \leftarrow 1/(1 + exp(s))$
    for all $f_i \in v_k$ **do**
        $w[f_i] \leftarrow w[f_i] - \alpha(p_k - y_k)/(\sqrt{n[f_i] + 1})$
        $n[f_i] \leftarrow n[f_i] + 1$
    end for
end for

▷ Testing the CTR model
for all $v_k \in testing$ **do** ▷ v_k is a *testing* sample
    $s \leftarrow 0$
    for all $f_i \in v_k$ **do**
        $s \leftarrow s + w[f_i + 1]$
    end for
    $p_k \leftarrow 1/(1 + exp(s))$
end for
Compute the accuracy of the model

Table 2: Fitness evaluation for each combination of probabilities of crossover and mutation. The reported fitness value is the average of the best evaluation during 30 experiments per combination.

p_m \ p_c	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
0.1	1306.48	1293.84	1285.14	1281.03	1325.25	1297.14	1316.91	1314.43	1288.35
0.2	1343.89	1314.14	1359.57	1303.31	1316.87	1317.57	1311.78	1322.47	1277.48
0.3	1313.68	**1375.71**	1348.36	1316.86	1309.21	1311.98	1315.48	1321.02	1299.37
0.4	1313.79	1328.77	1346.24	1319.09	1314.67	1312.18	1328.36	1316.44	1289.55
0.5	1337.88	1349.90	1337.83	1338.95	1336.79	1354.18	1289.94	1316.72	1303.00
0.6	1361.53	1354.09	1317.51	1324.24	1279.99	1316.87	1289.92	1289.59	1285.07
0.7	1345.76	1325.94	1341.92	1327.09	1309.07	1332.03	1287.41	1312.90	1249.44
0.8	1312.20	1332.25	1331.23	1339.68	1320.27	1300.19	1337.20	1288.46	1261.30
0.9	1351.96	1322.49	1326.18	1304.30	1319.22	1314.88	1278.75	1266.95	1278.97

we fixed the number of iterations to 300, for comparison purposes. As observed in Table 2, the best fitness evaluation was obtained when $p_c = 0.2$ and $p_m = 0.3$.

4.4. Description of the Experiments

For this work, we performed two experiments in order to determine the feasibility of the proposed online advertising campaigns optimization using genetic algorithms and the prediction CTR model, and also to determine the performance of the proposal for microtargeting. The details of the experiments are as follows:

I. *Optimization of Online Advertising Campaigns* – This experiment aims to implement the proposed optimization and to determine the characteristics of the optimization procedure. Algorithm 3 shows the implementation of our proposal by using the prediction CTR model depicted in Section 4.2 and the selected parameters analyzed in Section 4.3 with $T = 2000$. The testing dataset of 2-million samples was used. The analysis of the optimization procedure considered 50 runs of Algorithm 3 and the computation of the mean and standard deviation of the best fitness evaluation of each run.

II. *Optimization of Semi-Customized Advertising Campaigns* – This experiment aims to implement the proposed optimization with fixed features customized by the user, and to determine the characteristics of the optimization technique. In this regard, the following features, with their values, were fixed: $f_3 = 23$, $f_4 = 4101$, $f_5 = 0$ and $f_6 = 479741$. The other features should be optimized by our proposal. Notice that the values were extracted from the original dataset. Algorithm 3, with the prediction CTR model (Section 4.2) and the same parameters (Section 4.3) with $T = 2000$, was employed. The testing dataset of 2-million samples was used. The same analysis as in the previous experiment was considered.

All the experiments were implemented using R Studio version 3.3.2 environment and the GA package for R (Scrucca, 2013). We used a MacBook Pro 14.1 with OS Sierra 10.12.5, one 2.3 GHz Intel Core i5 processor with 2 cores, 8 GB at 2133 MHz LPDDR3 of RAM, L2 Cache (per core) of 256 KB, and L3 cache of 4 MB. For timing purposes, Experiment I was computed in $03h : 23m : 34s$ per repetition and Experiment II was computed in $00h : 24m : 05s$ per repetition.

5. Results and Discussion

This section reports the results from the experimentation explained above. Two experiments were conducted in order to determine the feasibility of the proposed GA-and-ML based optimization method and its usage for customized advertising campaigns.

5.1. Experiment I: Optimization of Online Advertising Campaigns

Firstly, the proposed online advertising campaigns optimization method was implemented as Algorithm 3. We ran the same algorithm 50 times in order to determine the characteristics of the optimization procedure. Figure 4 shows the results of this experiment, in which the strong line represents the mean, μ, of the best fitness evaluation at each generation, and the thin line the standard deviation, σ, of the best fitness evaluation at each generation. Particularly, Figure 4 shows the interval between $[\mu - \sigma, \mu + \sigma]$. It can be seen that 90% of the mean best fitness evaluation, i.e. $0.9\mu = 1170$, is reached in 20 generations. Also, the mean best fitness evaluation at the last generation (300) reaches a value of 1300 that represents 35% relatively less than the theoretical best fitness evaluation of 2000 (assuming that $T = 2000$ and the average CTR is 1.0). Additionally, the curve tends to increase while the generations are more. This behavior reflects that the proposed method based on genetic algorithms is well configured and implemented.

Algorithm 3 Proposed online advertising campaigns optimization.

Require: dataset, threshold T, prediction CTR model, GA parameters
Ensure: best campaign configuration

$\triangleright$ Set the parameters
$|population| \leftarrow 500$, $max\_iterations \leftarrow 300$, $p_c \leftarrow 0.2$, $p_m \leftarrow 0.3$ and elitism $\leftarrow 5\%$

$\triangleright$ GA begins
Generate an initial population
individuals $\leftarrow$ Generate allowed individuals $\triangleright$ see Figure 2
for i $\leftarrow 1 : max\_iterations$ **do**
  individuals $\leftarrow$ Verify or modify individuals to properly allow them
  for all *individuals* **do**
    Decode the individual, and set $chromosome_1$ and $chromosome_2$
    Select the columns of dataset where values in $chromosome_1$ are "1"
    Select those samples in dataset where values are equal to those of $chromosome_2$
    subset $\leftarrow$ Generate a subset using columns and samples $\triangleright$ see Figure 3
    if rows in $subset > T$ **then**
      $subset \leftarrow RandomSelection(subset, T)$ $\triangleright$ randomly select T samples
    end if
    $CTR_{average} \leftarrow$ Compute the average CTR of *subset* using (4)
    $D' \leftarrow size(subset)$
    $f(individual) \leftarrow CTR_{average} \times \min(D', T)$
  end for
  Select *individuals* using linear-rank selection and 5% elitism
  Crossover pairs of *individuals* with probability p_c using the single-point operator
  Mutate genes of *individuals* with probability p_m using uniform random operator
  individuals $\leftarrow$ new individuals
end for
Return best individual

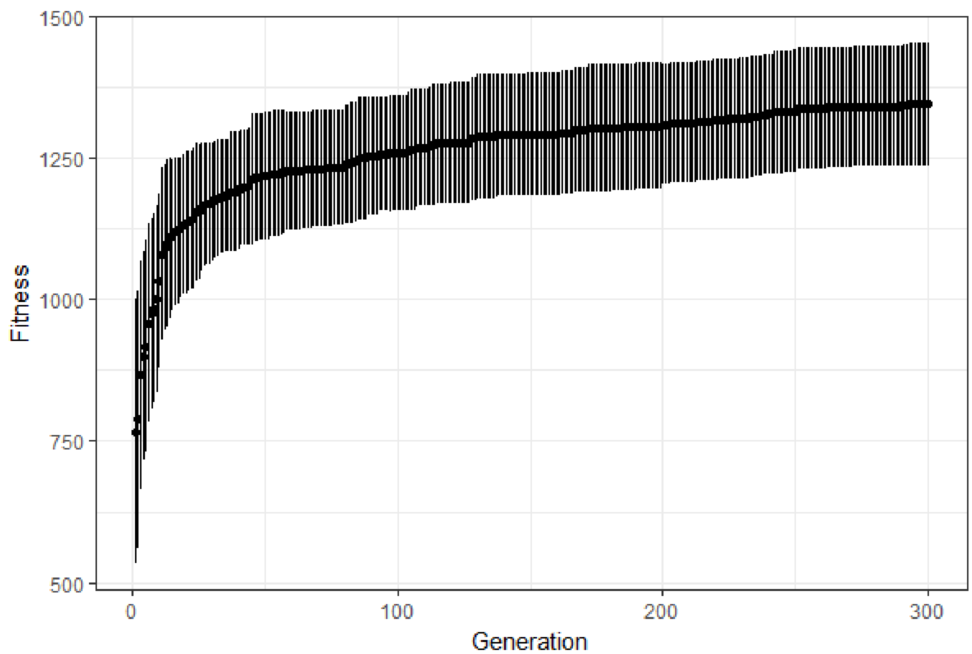

Figure 4: Results of the proposed online advertising campaigns optimization with 50 repetitions: (strong line) average μ of the best fitness evaluation at each generation, (thin lines) representation of the interval between the average and the standard deviation σ at each generation.

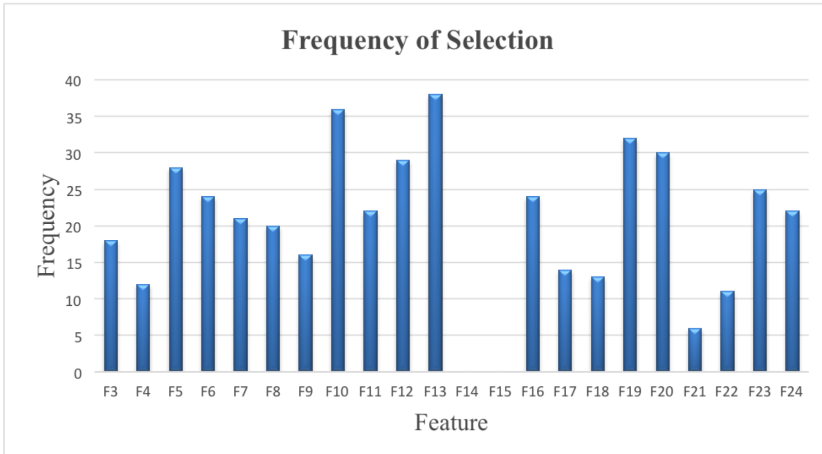

Figure 5: Bar graph representing the number of times each feature was selected by the best inviduals found with the proposed optimization method.

On the other hand, the best individual of each repetition was stored. Table 3 summarizes the best individual at each repetition sorted by the fitness function value. This table shows: the number of experiment, the iteration of the GA when the best fitness value were found, the number of features in the configuration, the average CTR of the subset selected, the real size of the whole subset before using the threshold value, the fitness function value, and the features selected by the individual.

In terms of results reported in Table 3, it is shown that in the first quartile: the number of features are 9, the average CTR value is 0.72, and the fitness function value is 1447.94. Also, Figure 5 shows the frequency of features selected during the 50 repetitions by the best individuals. The mean frequency of the features is 19.95, then features above it, i.e. f_i for all $i = \{5, 6, 7, 8, 10, 11, 12, 13, 16, 19, 20, 23\}$, were the most frequent ones considered by the best 50 individuals. These features are important since these configurations provide high evaluation which implies that the average CTR is larger as well as the size of the campaign. In addition, its is evident that features $i = \{14, 15\}$ were not considered, since those have high variance in their values generating too much small groups of ad campaigns. The latter impacts on the fitness evaluation, resulting in their lack of selection. To this end, the selected features suggest that the optimal advertising campaign should be configured with these requirements.

Table 3: Best individuals of the 50 repetitions in Experiment I. Individuals are sorted by fitness function value.

No.	Fitness	No. Exp.	Iteration	No. Features	Average CTR	Real Size	Features Configuration
1	1448.52	35	97	10	0.7243	2474	4, 5, 6, 7, 11, 12, 16, 19, 20, 23
2	1448.52	11	289	11	0.7243	2474	4, 5, 6, 7, 8, 10, 11, 12, 16, 20, 23
3	1448.40	10	289	7	0.7242	2474	6, 7, 10, 13, 16, 19, 23
4	1448.16	25	275	9	0.7241	2474	6, 8, 10, 11, 13, 16, 19, 20, 23
5	1447.86	12	61	11	0.7239	2447	4, 5, 6, 7, 10, 11, 13, 16, 17, 19, 23
6	1447.85	18	296	8	0.7239	2447	6, 8, 9, 10, 12, 17, 19, 23
7	1447.81	41	204	10	0.7239	2447	3, 7, 8, 9, 10, 12, 16, 17, 19, 23
8	1447.81	47	204	10	0.7239	2447	3, 7, 8, 9, 10, 12, 16, 17, 19, 23
9	1447.74	5	226	10	0.7239	2474	5, 7, 8, 9, 10, 11, 12, 13, 19, 23
10	1447.65	24	282	9	0.7238	2474	7, 8, 9, 10, 11, 12, 13, 20, 23
11	1447.57	21	199	9	0.7238	2474	3, 7, 8, 10, 11, 12, 13, 19, 23
12	1447.52	1	258	9	0.7238	2474	4, 6, 7, 9, 10, 13, 16, 19, 23
13	1447.51	4	138	9	0.7238	2474	4, 5, 7, 8, 12, 13, 19, 20, 23
14	1447.42	20	258	6	0.7237	2474	6, 7, 10, 13, 20, 23
15	1447.38	9	264	12	0.7237	2447	4, 5, 6, 7, 8, 10, 11, 13, 17, 19, 20, 23
16	1447.26	23	129	11	0.7236	2447	4, 5, 7, 9, 12, 13, 16, 17, 19, 20, 23
17	1446.92	29	295	10	0.7235	2447	5, 6, 7, 10, 11, 12, 13, 17, 20, 23
18	1445.92	30	293	12	0.7230	2447	3, 5, 7, 8, 10, 11, 12, 16, 17, 19, 20, 23
19	1440.29	2	248	11	0.7201	2059	4, 5, 7, 8, 9, 10, 13, 16, 19, 20, 22
20	1440.18	17	157	10	0.7201	2044	4, 7, 8, 9, 10, 11, 12, 17, 20, 22
21	1440.05	19	300	11	0.7200	2044	5, 6, 8, 9, 10, 12, 13, 16, 17, 19, 22
22	1439.72	42	245	5	0.7199	3476	7, 9, 16, 19, 20
23	1439.72	48	245	5	0.7199	3476	7, 9, 16, 19, 20
24	1438.97	13	293	9	0.7195	2044	3, 5, 6, 7, 16, 17, 19, 20, 22
25	1438.24	37	128	8	0.7191	3438	3, 5, 6, 8, 10, 12, 17, 19
26	1436.47	27	286	6	0.7182	3476	5, 6, 8, 9, 13, 20
27	1432.69	38	298	9	0.7163	3476	3, 8, 9, 10, 11, 13, 16, 19, 20
28	1429.12	3	293	11	0.7146	3476	3, 6, 7, 8, 9, 10, 11, 12, 16, 19, 20
29	1323.86	36	292	8	0.6619	4054	4, 5, 8, 10, 12, 13, 17, 22
30	1280.76	7	293	8	0.6404	4976	3, 5, 8, 9, 10, 12, 13, 24
31	1229.60	40	252	5	0.6148	2619	5, 10, 13, 20, 24
32	1229.60	46	252	5	0.6148	2619	5, 10, 13, 20, 24
33	1229.12	26	151	9	0.6146	2576	3, 5, 6, 10, 11, 13, 17, 20, 24
34	1227.97	16	288	6	0.6140	2619	9, 12, 13, 19, 20, 24
35	1223.21	8	202	8	0.6116	2029	3, 10, 11, 13, 18, 20, 22, 24
36	1223.06	15	48	11	0.6115	2028	4, 6, 10, 11, 12, 13, 16, 19, 22, 23, 24
37	1223.04	33	209	8	0.6115	2029	4, 5, 10, 12, 13, 21, 23, 24
38	1223.02	14	150	9	0.6115	2029	4, 5, 10, 13, 16, 18, 20, 22, 24
39	1222.96	28	232	10	0.6115	2028	5, 6, 10, 13, 16, 18, 20, 21, 22, 24
40	1222.94	22	178	12	0.6115	2028	3, 5, 6, 10, 12, 13, 16, 18, 19, 21, 23, 24
41	1222.90	44	70	7	0.6114	2029	5, 12, 13, 16, 18, 20, 24
42	1222.90	50	70	7	0.6114	2029	5, 12, 13, 16, 18, 20, 24
43	1222.85	43	53	9	0.6114	2028	3, 5, 6, 10, 11, 13, 18, 19, 24
44	1222.85	49	53	9	0.6114	2028	3, 5, 6, 10, 11, 13, 18, 19, 24
45	1222.80	39	117	9	0.6114	2029	8, 11, 12, 13, 16, 19, 21, 23, 24
46	1222.80	32	159	11	0.6114	2028	3, 6, 11, 13, 16, 19, 20, 21, 22, 23, 24
47	1222.79	6	213	8	0.6114	2029	3, 12, 13, 18, 19, 21, 22, 24
48	1222.58	34	96	7	0.6113	2029	6, 10, 13, 18, 20, 23, 24
49	1222.56	31	272	8	0.6113	2029	6, 12, 13, 18, 19, 22, 23, 24
50	1222.56	45	177	10	0.6113	2029	3, 5, 10, 11, 12, 13, 18, 19, 20, 24

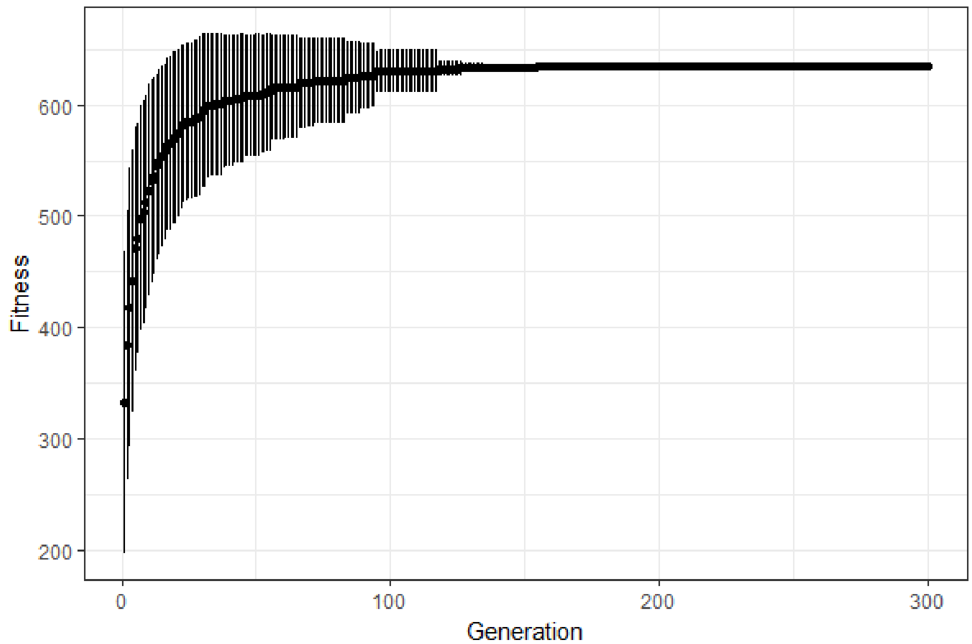

Figure 6: Results of the proposed optimization method with fixed features, during 50 repetitions: (strong line) average μ of the best fitness evaluation at each generation, (thin lines) representation of the interval between the average and the standard deviation σ at each generation.

5.2. Experiment II: Optimization of Semi-Customized Advertising Campaigns

Once the proposed online advertising campaings optimization method was proved. The next experiment considered to determine the characteristics of the optimization procedure when several features are initially fixed. This experiment aims to determine the feasibility of the proposed method to perform an optimization over a semi-customized advertising campaign approach in which the fixed features simulates the customization of some characteristics of the campaign by the advertiser. As already mentioned, features f_3, f_4, f_5 and f_6 were fixed (see Section 4.4).

Thus, we ran Algorithm 3 during 50 times. Figure 6 shows the results of this experiment, in which the strong line represents the mean, μ, of the best fitness evaluation at each generation, and the thin line the standard deviation, σ, of the best fitness evaluation at each generation.

As shown in Figure 6, the best individual reaches the mean value $\mu = 634$ after 155 generations, with standard deviation $\sigma \approx 0$. In that sense, it is evi-

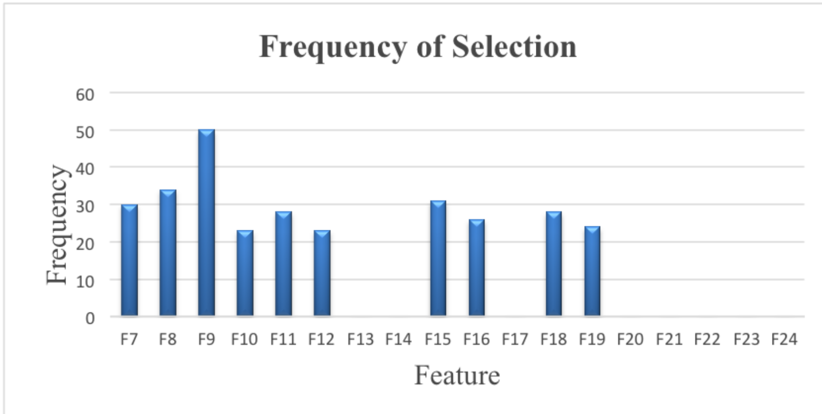

Figure 7: Bar graph representing the number of times each feature was selected by the best inviduals found with the proposed optimization method using fixed features.

dent that the maximum number of iterations ensures the optimal solution for this dataset. Furthermore, the best individual of each repetition was stored and depicted in Table 4 in ascending order by number of repetition, showing: the number of experiment, the iteration of the GA when the best fitness value were found, the number of features in the configuration, the average CTR of the subset selected, the real size of the whole subset before using the threshold value, the fitness function value, and the features selected by the individual. As noticed, the optimized configuration when some features are customized considers small average CTR (0.317) but large size of the campaign (2234 in average). This can be explained since some features are fixed, so the average CTR tends to decrease; but the proposed algorithm promotes also larger size values of campaigns.

Additionally, the most frequent features appeared in the best configurations over the entire experiment were calculated, as shown in Figure 7. It can be observed that the features f_i for all $i = \{7, 8, 9, 10, 11, 12, 15, 16, 18, 19\}$ were selected by the best individuals. Particularly, features f_8, f_9 and f_{15} were the three most frequent features selected. To this end, the selection of the latter features is justified by the fact that those combined with the given customized set of features should be an optimal online advertising campaign.

5.3. Discussion

Research has been done to distribute and optimize advertisers' budget, and microtargeting (see Section 2.2). However, to the best of our knowledge,

Table 4: Best individuals of the 50 repetitions in Experiment II. Individuals are sorted by fitness function value.

No.	Fitness	No. Exp.	Iteration	No. Features	Average CTR	Real Size	Features Configuration
1	635.08	27	172	6	0.3175	2121	7,9,11,15,16,18
2	634.94	15	92	7	0.3175	2348	7,8,9,10,11,15,18
3	634.91	25	127	4	0.3175	2348	8,9,12,19
4	634.91	38	226	7	0.3175	2348	7,8,9,11,12,18,19
5	634.89	36	155	3	0.3174	2348	7,8,9
6	634.86	22	232	4	0.3174	2348	8,9,15,18
7	634.77	10	100	6	0.3174	2121	7,9,11,12,16,18
8	634.70	49	167	5	0.3173	2121	8,9,10,11,16
9	634.66	37	138	6	0.3173	2348	8,9,11,15,18,19
10	634.65	14	197	4	0.3173	2348	7,8,9,11
11	634.61	48	122	5	0.3173	2348	7,9,10,11,19
12	634.60	3	143	8	0.3173	2121	7,9,10,11,12,15,16,19
13	634.55	4	106	7	0.3173	2348	7,8,9,11,12,15,18
14	634.54	12	73	6	0.3173	2121	7,9,11,12,15,16
15	634.54	41	193	9	0.3173	2121	8,9,10,11,12,15,16,18,19
16	634.53	32	46	6	0.3173	2121	7,8,9,11,15,16
17	634.51	18	55	5	0.3173	2348	8,9,10,12,15
18	634.51	44	79	7	0.3173	2348	7,8,9,12,15,18,19
19	634.50	6	70	4	0.3173	2348	7,8,9,18
20	634.50	30	114	5	0.3172	2121	7,9,11,12,16
21	634.50	47	183	4	0.3172	2348	8,9,11,15
22	634.49	43	208	7	0.3172	2121	8,9,10,15,16,18,19
23	634.43	21	166	5	0.3172	2348	7,8,9,12,15
24	634.39	17	162	8	0.3172	2121	7,8,9,11,15,16,18,19
25	634.38	34	161	4	0.3172	2121	7,9,10,16
26	634.30	45	96	7	0.3172	2121	7,8,9,12,15,16,19
27	634.29	33	78	7	0.3171	2121	7,9,11,12,15,16,18
28	634.29	39	162	7	0.3171	2348	7,9,11,12,15,18,19
29	634.24	28	195	7	0.3171	2121	7,9,10,11,15,16,19
30	634.23	2	129	5	0.3171	2348	7,9,11,18,19
31	634.22	11	176	4	0.3171	2348	7,9,15,18
32	634.18	19	89	6	0.3171	2121	8,9,10,16,18,19
33	634.18	31	161	4	0.3171	2348	7,9,10,12
34	634.18	16	191	8	0.3171	2348	7,8,9,10,12,15,18,19
35	634.15	46	56	7	0.3171	2121	8,9,10,15,16,18,19
36	634.14	42	285	6	0.3171	2121	7,8,9,15,16,18
37	634.08	29	25	8	0.3170	2121	8,9,10,12,15,16,18,19
38	634.07	26	184	6	0.3170	2121	8,9,11,12,16,19
39	634.06	9	61	8	0.3170	2121	8,9,10,11,15,16,18,19
40	634.02	8	246	6	0.3170	2348	8,9,10,11,12,18
41	634.01	24	74	6	0.3170	2121	8,9,10,12,16,18
42	633.99	40	116	8	0.3170	2121	7,9,10,11,15,16,18,19
43	633.98	20	63	4	0.3170	2348	8,9,10,15
44	633.97	1	90	6	0.3170	2348	7,9,11,15,18,19
45	633.95	23	186	4	0.3170	2348	8,9,10,11
46	633.89	50	70	5	0.3169	2348	8,9,12,18,19
47	633.85	7	59	5	0.3169	2121	8,9,10,15,16
48	633.84	13	133	8	0.3169	2121	7,8,9,11,15,16,18,19
49	633.82	5	249	7	0.3169	2121	7,9,10,12,15,16,19
50	633.65	35	112	6	0.3168	2121	8,9,10,11,12,16

no research has been published that focus to benefits small ad networks. Our proposed methodology can be easily implemented and at a very low cost, which is very beneficial for small advertising networks. Once advertisers configure their campaigns, the option of automatic configuration could be offered to them. Automatic configuration facilitates advertisers the launch of campaigns and increases its profitability, which means having more satisfied advertisers. Best configurations could be obtained offline. This is, the algorithm could be running all the time, and store best solutions on a list. Once a user configures a campaign some optimized campaigns could be offered to him/her by consulting the list.

Our methodology could be easily reconfigured by changing the fitness function. For example, if advertisers seek to increase their products sales or the number of forms filled by users, instead of using the average CTR parameter, the likelihood of generating a sale or that a form is filled out could be used.

Since the spectrum of advertisers ranges from small entrepreneurs starting their business to multinational companies that make massive campaigns in many countries, we consider that each advertiser needs a different number of visits to meet their campaigns. For this reason, the threshold should be related to each advertiser budget.

6. Conclusions and Future Work

This article presents a novel methodology to find optimal campaign settings for advertisers in a automatic way. We consider the methodology to be very useful for small advertising networks because they can optimize advertisers campaigns effectively using few resources. In this way, small networks may be more competitive and may have more satisfied advertisers. Our methodology can be applied very easily to real-time bidding (RTB) that is a huge auction involving many ad networks. This improvement can make RTB a much more attractive system to advertisers.

Supervised ML models enable predicting CTR, in such a way that it is not no longer necessary that advertisers invest in campaigns before getting an optimal configuration. Since it is possible to simulate users behavior artificially, better suggestions could be made to advertisers in less time and at little expense, which may increase their satisfaction degree.

For the methodology to be effective, it is required a CTR model as precise as possible. Thus, improving the CTR precision is one of the lines of improvement. Since new browsers, new operating systems and new devices constantly appear, the generated traffic on the Internet is constantly changing. Thus, it is necessary to update the CTR model frequently. In addition, a larger dataset increases the search space and better solutions could be found, although better computation resources would be required.

Other versions of crossover and mutation operations, as well as the way first population is generated could also be tested. For this purpose, many more experiments should be made. For example, instead of generating randomly all individuals in the first population, many of them could be obtained from the best solution list.

To conclude, our methodology could also be improved by running it in a shorter time. Since a number N of experiments can be executed in parallel, the algorithm is easily scalable. To do this, the dataset can be replicated N times, then parallel executing many instances of the algorithm and add solutions to a common list. The more instances are executed, the greater likelihood of obtaining better solutions. In addition, in order to optimize time all individuals fitness could be saved on a shared list, in such a way that before calculating the fitness the list is consulted. This could be save a lot of time in the long term.

References

Aronowich, M., Benis, A. J., Yanai, R., Vind, G., Jun. 25 2014. Budget distribution in online advertising. US Patent App. 14/314,151.

Avazu, 2015.
URL `http://www.kaggle.com/c/avazu-ctr-prediction/data`

Balseiro, S. R., Feldman, J., Mirrokni, V., Muthukrishnan, S., 2014. Yield optimization of display advertising with ad exchange. Management Science 60 (12), 2886–2907.

Chapelle, O., Manavoglu, E., Rosales, R., 2015. Simple and scalable response prediction for display advertising. ACM Transactions on Intelligent Systems and Technology (TIST) 5 (4), 61.

Chen, G., Cox, J. H., Uluagac, A. S., Copeland, J. A., 2016. In-depth survey of digital advertising technologies. IEEE Communications Surveys & Tutorials 18 (3), 2124–2148.

Chin, W.-S., Zhuang, Y., Juan, Y.-C., Lin, C.-J., 2015. A learning-rate schedule for stochastic gradient methods to matrix factorization. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 442–455.

Evans, D. S., 2008. The economics of the online advertising industry. Review of network economics 7 (3).

Evans, D. S., 2009. The online advertising industry: Economics, evolution, and privacy. The journal of economic perspectives 23 (3), 37–60.

Goldfarb, A., 2014. What is different about online advertising? Review of Industrial Organization 44 (2), 115–129.

Goldfarb, A., Tucker, C., 2011. Online display advertising: Targeting and obtrusiveness. Marketing Science 30 (3), 389–404.

IAB, Jun. 25 2016. Interactive advertising bureau. Inform of IAB.

Lee, Kuang-chih, Orten, B., Dasdan, A., Wentong, L., 2012. Estimating conversion rate in display advertising from past performance data. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 768–776.

Lee, K.-C., Jalali, A., Dasdan, A., 2013. Real time bid optimization with smooth budget delivery in online advertising. In: Proceedings of the Seventh International Workshop on Data Mining for Online Advertising. ACM, p. 1.

Li, P., Shrivastava, A., Moore, J. L., König, A. C., 2011. Hashing algorithms for large-scale learning. In: Advances in neural information processing systems. pp. 2672–2680.

McMahan, H. B., Holt, G., Sculley, D., Young, M., Ebner, D., Grady, J., Nie, L., Phillips, T., Davydov, E., Golovin, D., et al., 2013. Ad click prediction: a view from the trenches. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 1222–1230.

Miralles-Pechuán, L., Rosso, D., Jiménez, F., García, J. M., 2017. A methodology based on deep learning for advert value calculation in cpm, cpc and cpa networks. Soft Computing 21 (3), 651–665.

Muthukrishnan, S., Pál, M., Svitkina, Z., 2007. Stochastic models for budget optimization in search-based advertising. In: International Workshop on Web and Internet Economics. Springer, pp. 131–142.

Perlich, C., Dalessandro, B., Hook, R., Stitelman, O., Raeder, T., Provost, F., 2012. Bid optimizing and inventory scoring in targeted online advertising. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 804–812.

Provost, F., Dalessandro, B., Hook, R., Zhang, X., Murray, A., 2009. Audience selection for on-line brand advertising: privacy-friendly social network targeting. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp. 707–716.

Richardson, M., Dominowska, E., Ragno, R., 2007. Predicting clicks: estimating the click-through rate for new ads. In: Proceedings of the 16th international conference on World Wide Web. ACM, pp. 521–530.

Scrucca, L., 2013. GA: a package for genetic algorithms in R. Journal of Statistical Software 53 (4), 1 – 37.

Shan, L., Lin, L., Sun, C., Wang, X., 2016. Predicting ad click-through rates via feature-based fully coupled interaction tensor factorization. Electronic Commerce Research and Applications 16 (2016), 30–42.

Sivadas, E., Grewal, R., Kellaris, J., 1998. The internet as a micro marketing tool: targeting consumers through preferences revealed in music newsgroup usage. Journal of Business Research 41 (3), 179–186.

Wilbur, K. C., Zhu, Y., 2009. Click fraud. Marketing Science 28 (2), 293–308.