

Population-based Method for Optimizing Targeted Offers Problem in Direct Marketing Campaigns

MOULAY YOUSSEF SMAILI
University Ibn Tofail-National School of
Applied Science-Kenitra
Laboratory: System of Engineering
Kenitra-MOROCCO
Smailli.moulayyoussef@uit.ac.ma

HANAA HACHIMI
University Ibn Tofail-National School of
Applied Science-Kenitra
Laboratory: System of Engineering
Kenitra-MOROCCO
University Soultane Moulay Slimane – Beni
Mellal
Email: hanaa.hachimi@univibntofail.ac.ma

Abstract: - In direct marketing campaigns, the optimization of targeted offers problem is a big business concern. The main goal is to maximize the company's profit by reaching the right clients. The main challenge faced by companies when advertising, is to configure properly a campaign by choosing the appropriate target, so it is guaranteed a high acceptance of users to advertisements. When dealing with an important size of data, the important specification to consider is the combinatorial aspect of the problem and the limitation of the approach based on mathematical programming methods. In this article, and since this problem belongs to the class of

NP -hard problems, the use of metaheuristic, instead of exact methods, is essential; the Bat Algorithm which is a new inspired algorithm is proposed after hybridization with Genetic Algorithm. Computational experiments show that the proposed algorithm was able to give good and competitive solutions.

Key-Words: - Bat Algorithm, Discrete Optimization, Hybridization, Target Marketing.

Received: February 15, 2020. Revised: July 4, 2020. Accepted: August 10, 2020. Published: September 17, 2020.

1 Introduction

There are two axes which characterize research in metaheuristic: exploration and exploitation. Exploration guarantees a diversified search in the search space regions; however, the exploitation seeks to ensure the intensification of the search in order to find a better solution. The goal is to have a balance between these two contradictory objectives [6]. In the literature, many works make a difference between local search methods, instead of single solution-based and population-based methods. In this work, and unlike previous works, we deal with the targeted offers problem in direct marketing campaigns using population-based methods, and we focus on two algorithms; the Genetic Algorithm which can easily escape from the local minimum, and the Bat Algorithm (BA) which has the characteristic to quickly converge towards a local solution. Based on the concept of mimicking the mechanism of natural choice and heredity, the Genetic Algorithm is considered one of the most famous evolutionary search techniques. It has been applied to tackle a set of optimization problems where a deterministic polynomial solution is infeasible. The Bat Algorithm, inspired by the

foraging behavior of micro-bats, has been developed by Yang in 2010 [9], and it can be considered as an effort to gather positive side of previous metaheuristic algorithms. Thanks to its simplicity, flexibility, and easy implementation, the Bat Algorithm works well with complicated problems. The main feature of this algorithm is the quick convergence at a very initial stage by keeping a balance when switching from exploration to exploitation. In order to maximize their benefits, firms use "marketing campaigns" as a fundamental tool, along two main axes; by acquiring new customers, so-called acquisition or by generating additional revenues with their existing customers by offering new products to them; and this is called retention. This paper focuses on the second axis (retention) in order to maximize the marketing return on investment while choosing which product to offer, and to which customer, under a set of constraints by using a new Hybrid Bat Algorithm with Genetic Algorithm. The final goal is to choose the right product, to offer to the right customer, at the right time to maximize the marketing Return on Investment (ROI). This goal is not easy to implement because of the fact that firms have

multiple products to offer under a complex set of business constraints [1]. In this work, and due to the concrete aspect of this kind of problems, the use of a binary version of the Bat Algorithm is essential. The standard BA was originally designed for the optimization of non-discrete problems [4]. The objective is to find a solution with a discrete value. The sought solution is represented by a matrix which contains binary values; 1 if product j is offered to customer 1, and 0 otherwise. This paper is organized as follows, section 2 describes some related works in the same area; works about the targeted marketing problem are mentioned first, followed by works which process the Bat Algorithm's binary aspect in a discrete search space. In section 3, we describe the problem of Market Targeting and the modelling adopted to search a solution for the problem. In section 4, we describe two metaheuristic algorithms; namely the Binary Bat Algorithm (BBA), and the Genetic Algorithm. In this section we present our proposed algorithm: Hybridized Genetic Binary Bat Algorithm (HGBBA) developed to encounter the shortcomings of the two first algorithms while searching for adapted solutions. Experimental results are presented in section 5 for the three algorithms with a comparison of their performance.

Related works - The approach of Cohen et al. [1] was to transform the database marketing problem into an optimization problem intended to generate the maximum profit under a set of business constraints. Approximately 2.5 million customers, eleven unique offers, and five investments are considered. The solution provided an approximately optimal solution to the ideal capacity assignment problem. The output is a decision, for each customer as to which, if any, product to offer and through what channel. Nobibon et al. [5] presented eight algorithms to solve the problem of maximizing the Return on Investment of targeted marketing. Seven of these algorithms are considered as exact methods and the eighth one is a heuristic method: the Tabu Search (TS). In the results, the TS showed the best performances. Oliveira et al. [8] took up the problem as described by Nobibon et al. [5], focusing on solving the problem by developing an efficient technique. The approach followed by Oliveira et al. [8] consists of hybridizing two metaheuristics: Greedy Randomized Adaptive Search Procedure (GRASP), and General Variable Neighborhood Search (GVNS). Three steps characterize the method followed by Oliveira et al. [8]; the first one verified the efficiency of the GRASP algorithm proposed to initialize a good solution. In the second

step, Oliveira et al. [8] kept the same parameters, but this time, they called the VND procedure for a local search. In the third step the complete proposed algorithm GGVNS was tested and it provided better results. Victor et al. [11] proposed a multi-objective DMLS metaheuristic for searching for sets of non-dominated solutions, in order to maximize both the promotion campaign's total profit and the risk-adjusted return. In order to remedy to the concept of volatility, Vector et al. [11] proposed a greedy randomized initial solution builder, able to perform local searches considering different neighbourhood exploration techniques, and a genetic method is designed. All the works cited above used metaheuristic based on local search methods to ensure an optimization of the return on investment, while respecting the business constraints. On the other side, the use of a Binary Bat Algorithm will allow us to deal with the target marketing optimization problem by looking for solutions made up of discrete (binary) values instead of looking for continuous solutions. Hence, in designing the binary version of the Bat Algorithm, some main concepts of the velocity and position updating process must be adapted in order to guarantee the binary aspect of the research. For Seyedali et al. [7], the objective was to find how the concept of velocity in a real space should be employed in order to update the positions in a binary space. The idea was to change the position of micro-bats with the probability of their velocity. To do that, a transfer function is used to map velocity values to probability values while updating the position. In other words, a transfer function calculates the probability of changing a position from 0 to 1 and vice versa.

2 Problem Description

Direct Marketing is a tool that allows firms to promote their products directly to customers, and measure results quickly. One of the most important benefits of Direct Marketing is "Upgrading firm's loyalty strategies" in order to maximize the companies' Return on Investment. Nobibon et al. [5], and Cohen [1] before, presented the formulation of the product targeting problem as a mixed-integer programming (MIP) problem including more business constraints, and it can be rewritten as follows: Given a set of m customers $C = [c_1, c_2, \dots, c_m]$, and a set of n products' offers $P = [p_1, p_2, \dots, p_n]$, the objective is to maximize the Return on Investment under these business constraints:

- The corporate hurdle rate: each company defines its hurdle rate (HR) to make sure that the Return on Investment is equal, at least, to a value of HR.

- During the campaign, the budget of each product is limited.
- A limitation is imposed on the total number of products offered to each customer.
- And there is also a MQC, which means that no customer will receive an offer of a product which is not part of the campaign. If the product belongs to the proposed ones then at least $P_j > 0$ clients will receive an offer.

A solution is represented by a binary array $R_{|C| \times |P|}$, where C indicates the set of available costumers, and P represents the possible products to be used in the campaign. If a given cell $s_{ij} |i \in C, j \in P$ is equal to "1" (true), the product j will be offered to the client i ; otherwise, the value would be "0" (false). There are two basic parameters of the customer lifetime value's computation; p_{ij} is the probability that client i responds positively to an offer of product j , and DFV_{ij} is the return to the firm when client i responds positively to the offer of product j [5, 4]. A basic formulation for the product targeting problem can be expressed as:

$$\sum_{i=1}^m \sum_{j=1}^n (r_{ij} - c_{ij})x_{ij} - \sum_{j=1}^m fc_j y_j \quad (1)$$

Where:

- r_{ij} is the expected revenue of the company from the offer of product "j" to client "i", and $r_{ij} = p_{ij} DFV_{ij}$
- c_{ij} is the cost associated with the offer of product "j" to client "i"
- M_i is the upper bound of products to offer to a customer "i"
- P_j is the minimum quantity commitment bound associated with product "j"
- B_j is the budget, in the campaign, allocated to the product "j"
- fc_j is the cost needed if a product j is used in the campaign
- HR is the hurdle rate specific to each company

The goal is to maximize the evaluation function given by Eq. (1), by finding the optimal combination of the two matrices X and Y :

$$X = \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \end{pmatrix} \text{ and } Y = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

Where:

$$x_{ij} = \begin{cases} 1 & \text{If product } j \text{ is offered to customer } i, \\ 0 & \text{Otherwise} \end{cases}$$

and,

$$y_j = \begin{cases} 1 & \text{If product } j \text{ is used in the campaign,} \\ 0 & \text{Otherwise} \end{cases}$$

The constraints, to which the problem is subject to, can be modelled as follows:

$$1. \sum_{i=1}^m \sum_{j=1}^n r_{ij} x_{ij} \geq (1+HR) \left[\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n fc_j y_j \right] \quad (2)$$

$$2. \sum_{i=1}^m c_{ij} x_{ij} \leq B_j \quad \text{for each } j = 1 \dots, m \quad (3)$$

$$3. \sum_{j=1}^n x_{ij} \leq M_i \quad \text{for each } i = 1 \dots, m \quad (4)$$

$$4. \sum_{i=1}^m x_{ij} \leq m y_j \quad \text{for each } j = 1 \dots, n \quad (5)$$

$$5. \sum_{i=1}^m x_{ij} \geq P_j y_j \quad \text{for each } j = 1 \dots, n \quad (6)$$

$$6. x_{ij}, y_{ij} \in \{0, 1\} \quad \text{for each } i=1 \dots m \text{ and } j=1 \dots, n \quad (7)$$

Figure 1 below, illustrates an example of a problem with an advertising campaign optimization. The matrix represents a solution $S_{|C| \times |P|}$ concerning 5 products to offer to 10 customers, so as to optimize the Return on Investment by taking into account business constraints (maximum products to offer for each customer, fixed cost and budget for each product). In summary, this figure represents the offers to be adopted by each client. The first column on the left, represents the 10 customers who are the subject of the campaign with the maximum product values to offer to each

Customer. The first line at the top represents the different product offers with the fixed cost of each product and the budget allocated during the campaign. the values 1 means that product j is offered to customer i during the campaign and 0 if not.

	P_1 $f_{c_1}=16$ $R_{i=10}$	P_2 $f_{c_2}=51$ $R_{i=8}$	P_3 $f_{c_3}=46$ $R_{i=9}$	P_4 $f_{c_4}=36$ $R_{i=7}$	P_5 $f_{c_5}=32$ $R_{i=11}$
C_1 $M_{i=3}$	1	0	0	1	1
C_2 $M_{i=3}$	0	0	1	0	0
C_3 $M_{i=2}$	1	1	0	0	1
C_4 $M_{i=2}$	1	0	1	0	1
C_5 $M_{i=2}$	0	1	0	0	0
C_6 $M_{i=2}$	1	0	0	1	0
C_7 $M_{i=3}$	0	1	1	0	1
C_8 $M_{i=2}$	1	0	1	0	0
C_9 $M_{i=2}$	1	0	0	0	1
C_{10} $M_{i=3}$	1	0	1	0	1

Figure 1 : Example of Targeted Offers problem in Direct Marketing

3 Methodology

3.1 Building an initial solution with the Genetic Algorithm (GA)

One of the very important phases when dealing with the targeted offers problem in direct marketing campaigns, is the creation of an initial good quality solution which must be feasible by respecting the business constraints. The classical initialization of the Bat Algorithm, and which is based on randomization, creates a non-feasible or poor-quality initial solution. To solve this problem, we opted to build the initial solution using GA with a reduced population, and a number of reduced chromosomes. Genetic Algorithms are considered as the most known heuristic technique of search and optimization, and are inspired by natural evolution. They have proven to be effective in several real-world problems of significant complexity [2]. GA modeled the decision variable of a search problem by a string composed of alphabets with a certain cardinality. Each string is considered as a candidate solution to the search problem, and it is referred to as chromosome, while the alphabets composing the chromosomes are referred to as genes. In the problem of marketing campaigns' optimization, a chromosome is considered as a string representation of solutions to the problem. In this kind of problems, the classical GA is imposed and uses a bit-string representation to encode solutions. Bit-string chromosomes consist of a string of genes whose allele values are characters from the alphabet $\{0,1\}$; 1 if a product j is offered to customer i , and 0

if not. The gene may represent a decision to offer a product j to a specific customer i . The final goal is to maximize the fitness function. It's a computation that evaluates the quality of the chromosome as a solution to a particular problem. By analogy with biology, the chromosome is referred to as the genotype, whereas the solution it represents is known as the phenotype [2]. Genetic Algorithm is also characterized by the notion of population; Each GA operates on a population of artificial chromosomes; these are strings in a finite alphabet (usually binary). GA is composed of 6 steps to find an optimal solution for the problem; the objective in targeted offers problem is to maximize the return on investment by finding an optimal solution of equation (1):

1. Step 1: Initialization

A randomized procedure is usually used when generating the initial population of the candidate solution. However, if we randomly choose the composition of the chromosomes with a weighting of 50% for each value 0 and 1, we will obtain a non-feasible solution which do not respect the business constraints. For this, we have opted in this paper for a weighting distribution of 80% when a product j is offered to customer i , and 20% when it is not. With this method we have the certainty that our initial solution respects the business constraints but it will not necessarily represent an initial good quality solution.

2. Step 2: Evaluation

Once the population is generated (population initialized), the fitness values of the candidate solutions are evaluated to determine how fit an individual is.

3. Step 3: Selection

The selection is a principle that prefers better solutions over the worst ones, by allocating copies of those solutions with the highest fitness values; in this way the survival-of-the-fittest mechanism is imposed on candidate solutions.

4. Step 4: Crossover

Recombination combines parts of many parental solutions to create better solutions. Crossover is the most significant phase in a genetic algorithm.

5. Step 5: Mutation

The mutation occurs to maintain diversity of the population and escape premature convergence, by modifying the solution in an arbitrary way while acting locally.

6. Step 6: Termination

The algorithm terminates if there is no improvement in the solution quality, when time is reached, or when an acceptable solution is obtained.

The following is a pseudo code of the Standard Genetic Algorithm (GA):

Algorithm 1: Genetic Algorithm

- 1: Generate initial population randomly
 - 2: Evaluate the fitness of each individual in the population
 - 3: **for** i = 1 to MaxGen
 - 4: Select the best-fit individuals for reproductions
 - 5: Breed new individuals through crossover and mutation operations to give birth to offspring
 - 6: Evaluate the individual fitness of new individuals
 - 7: Replace least-fit population with new individuals
 - 8: Return the best solution in the population
-

Figure 2 Pseudo Code of Genetic Algorithm

3.2 Binary Bat Algorithm (BBA)

Bat Algorithm (BA) is one of the recently proposed heuristic algorithms imitating the echolocation behavior of micro-bats to perform global optimization. The superior performance of this algorithm has been proven among the other most well-known algorithms, such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) [7,14]. However, the classical Bat Algorithm does not generate discrete, but rather continuous solutions. The problem is to find solutions with discrete values (1 if product j is offered to customer i and 0 if not). For that reason, in this paper, the use of a binary version of Bat Algorithm is imposed [4]. In order to adapt this algorithm to the context of the Targeted Offers problem in Direct Marketing Campaigns (TODMC), a binary version of Bat Algorithm will be used and which is proposed by Seyedali et al. [7,14]. This version is called Binary Bat Algorithm (BBA). By modifying the micro-bats positions values from “0” to “1” and vice versa, the BBA moves micro-bats in a binary search space [7, 14]. The two main characteristics of micro-bats when finding their prey have been adopted in designing the Bat Algorithm. Micro-bats tend to decrease the loudness, and increase the rate of emitted ultrasonic sound when they chase prey. This behavior has been mathematically modeled as follows [9]: Each micro-bat i is characterized by its position X_i^t , and its velocity V_i^t at a specific iteration (time). Generating new solutions is performed by moving virtual micro-bats according to the following equations:

$$v_i^{t+1} = v_i^t + (x_i^t - G^{best})Freq_i \quad (8)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (9)$$

Where G^{best} is the best solution attained so far, and $Freq_i$ indicates the frequency of the i-th microbat which is calculated as follows:

$$Freq_i = Freq_{min} + (Freq_{max} - Freq_{min})\beta \quad (10)$$

Where β is randomly generated from a uniform distribution in [0,1]. Thanks to these equations the exploitability aspect is ensured in the Bat Algorithm. However, the exploitation side is guaranteed by a random walk procedure as follows:

$$x_{new} = x_{old} + \varepsilon L^t \quad (11)$$

L^t is the loudness of the bats' emitted sound, in order to perform an exploration instead of exploitation. L^t is the average loudness of all the micro-bats at this time step: $L^t = \langle L_i^t \rangle$, and ε is a random number in [-1,1] [9,7,14].

In order to achieve the balance between exploration and exploitation during the search process, loudness L^t , and pulse emission rate r_i should be updated only if the candidate solution is improved as the iterations proceed. They are updated by the following equations [9,14]:

$$L_i^t = \alpha L_i^t \quad (12)$$

$$r_i^t = r_i^0 [1 - \exp(-\gamma^t)] \quad (13)$$

Where α and γ are predefined constants. The search principal in BA, is once a micro-bat has found the prey's location, the loudness will be decreased while the pulse rate will be increased. However, The Bat Algorithm is designed, at the origin, to search in a continuous space of solutions. The algorithm, in its original version is not applicable when the set of solutions is discrete, because in a binary space, due to dealing with only two numbers ("0" and "1"), the position updating process cannot be performed using Eq. (9). To encounter this obstacle, Seyedali et al. [7, 14] proposed a solution: the idea is to use a transfer function with the aim of updating the position of micro-bats with the probability of their velocities. In practice the transfer function has the role of mapping velocity values to probability values. Needless to say, transfer functions force particles to move in a binary space. The most important factor for designing this function is to force micro-bats with high velocities to switch their positions. To do this, a v-shaped transfer function and position updating rule are proposed in

[7,14], and described in Equations (14) and (15) as follows:

$$V(v_i^k(t)) = \left| \frac{2}{\pi} \arctan \left(\frac{\pi}{2} v_i^k(t) \right) \right| \quad (14)$$

$$x_i^k(t+1) = \begin{cases} (x_i^k(t))^{-1} & \text{if rand} < V(v_i^k(t+1)) \\ x_i^k(t) & \text{if rand} \geq V(v_i^k(t+1)) \end{cases} \quad (15)$$

$x_i^k(t)$ and $v_i^k(t)$ are the position and velocity of microbat "i" at the iteration "t", where k is the problem dimension. $(x_i^k(t))^{-1}$ is the complement of $x_i^k(t)$. Figure 3 presents the general steps of the Bat Algorithm in its binary version.

Algorithm 2: Bat Algorithm

- 1: Initialize the bat population: X_i ($i = 1, 2, \dots, n$) = rand(0 or 1) and $V_i = 0$
 - 2: Define pulse frequency $Freq_i$
 - 3: Initialize pulse rates r_i and the loudness L_i
 - 4: **While** (num_iteration < Max generation iterations)
 - 5: Update frequency and velocities
 - 6: Calculate transfer function value using equation (14)
 - 7: Update micro-bats' positions by executing equation (15)
 - 8: **if** (rand > r_i)
 - 9: Select a solution (G^{best}) among the best solutions randomly
 - 10: Change some of the dimensions of position vector with some of G^{best} dimensions
 - 11: **end if**
 - 12: Generate a new solution by flying randomly
 - 13: **if** (rand < L_i and $f(x_i) > f(G^{best})$ Constr(G^{best}) = True)
 - 14: Accept the new solutions
 - 15: **end if**
 - 16: Evaluate the fitness of each individual in the population
 - 17: Increase r_i and reduce L_i
 - 18: Rank the micro-bats and find the current G^{best}
 - 19: **end while**
 - 20: Return the best solution.
-

Figure 3 Pseudo Code of Standard Bat Algorithm

3.3 Proposed Hybrid Approach

While dealing with the problem of targeted marketing maximization, we note that the Genetic Algorithm takes a considerable time to find a solution but with a minimum chance of blocking in a local minimum, while the Bat Algorithm has the advantage of quickly converging on a solution to the problem. However, the Bat Algorithm can easily be trapped in a local minimum. Taking into account the random aspect of the initialization procedures, the two

algorithms suffer from the poor quality of the initial solution. To overcome these obstacles, we propose to hybridize the Binary Bat Algorithm, to benefit from convergence in advanced iterations, with the Genetic Algorithm that generates better population from good parents. These results are close to a global optimum. The initial solution in the proposed algorithm is generated by performing the Genetic Algorithm for the first time, with a reduced number of chromosomal parents; this will quickly generate an initial solution

of better quality than standard algorithms. The exploration and exploitation are controlled by pulse emission rate, in the case of standard Binary Bat Algorithm [10,15]. This factor increases as the iteration proceeds, and the algorithm gradually loses exploitation capability; by inserting linear decreasing

inertia weight factor, the exploitation capability of the algorithm is improved while the exploration capability will be improved by the hybridization of both algorithms. Figure 4 presents the steps of the proposed algorithm:

Algorithm3 : HGBBA

```

1:
2: Executing GA to initialize micro-bats population:  $X_i$  ( $i = 1, 2, \dots, n$ )
3:   for  $i = 1$  to  $\text{MaxGen}(\text{GA})$ 
4:     Select the best-fit individuals for reproductions
5:     Breed new individuals through crossover and mutation operations to give birth to offspring
6:     Evaluate the individual fitness of new individuals
7:     Replace least-fit population with new individuals
8:     Return the best solution in the population
9:   if ( $f(\text{BestChromosom}) > f^{(\text{G}^{\text{best}})}$ )
10:    accept BestChromosom as a new solution
11:   end if
12: Define pulse frequency  $\text{Freq}_i$ 
13: Initialize pulse rates  $r_i$  and the loudness  $L_i$ 
14: while ( $\text{num\_iteration} < \text{Max generation iterations}$ )
15:   Update frequency and velocities
16:   Calculate transfer function value using equation (14)
17:   Update micro-bats positions by executing equation (15)
18:   if ( $\text{rand} > r_i$ )
19:     Select a solution ( $\text{G}^{\text{best}}$ ) among the best solutions randomly
20:     Change some of the position vector dimensions with some of the  $\text{G}^{\text{best}}$  dimensions
21:   end if
22:   Generate a new solution by flying randomly
23:   if ( $\text{rand} < L_i$  and  $f(x_i) > f(\text{G}^{\text{best}})$  and  $\text{Constr}(\text{G}^{\text{best}})=\text{True}$ )
24:     Accept the new solutions
25:   end if
26:   Initialize GA by the new population generated by BBA in an iteration “t”
27:   Evaluate the fitness of each individual in the population
28:   For  $i = 1$  to  $\text{MaxGen}(\text{GA})$ 
29:     Select the best-fit individuals for reproductions
30:     Bread new individuals through crossover and mutation operations to give birth to offspring
31:     Evaluate the individual fitness of new individuals
32:     Replace least-fit population with new individuals
33:     Return the best solution in the population
34:     Evaluate the fitness of each individual in the population
35:   Increase  $r_i$  and reduce  $L_i$ 
36:   Rank the micro-bats and find the current  $\text{G}^{\text{best}}$ 
37: end while
38: Return the best solution
    
```

Figure 2 Pseudo Code of Hybridized Genetic Binary BA (HGBBA)

4 Computational experiments and discussions

The three algorithms have been coded in MATLAB; the experiments were run on a Lenovo TS470 personal computer Core i7, with 2.6 GHz clock speed and 16 GB RAM, on a Windows 10. CPLEX 12.10 was used to get the linear solution. The dataset used to test and compare

the three algorithms is generated in a random manner described by Nobibon et al. [5]:

- Cost c_{ij} is randomly generated from the set $\{1,2,3\}$.
- The return of the firm r_{ij} is an integer generated randomly between 0 and 16.
- The corporate hurdle rate HR can take 5%, 10%, and 15% values.

- In this paper, there are three different values of the number of customers: 100 customers for small category S, 1000 customers for medium category M, and 10,000 customers for large category L.
- For each category of customers, we have two different numbers of product n: 5 and 10 products.
- For each combination (category of customer and number of products), a minimum-quantity commitment bound P_j is randomly generated between $\left\lfloor \frac{\sum_i M_i}{n} \right\rfloor$ and $\left\lfloor 2 \frac{\sum_i M_i}{n} \right\rfloor$.
- The fixed cost fc_j is generated randomly between $\frac{O_j}{2m(1+HR)} \sum_i [p_{ij} - (1+HR)c_{ij}]$ and $\frac{O_j}{m(1+HR)} \sum_i [p_{ij} - (1+HR)c_{ij}]$
- For budget B_j , three values are adopted: the two values of $\left\lfloor O_j \frac{\sum_i c_{ij}}{m} \right\rfloor$ $\left\lfloor 2 \frac{\sum_i M_i}{n} \frac{\sum_i c_{ij}}{m} \right\rfloor$ and a random integer between $\left\lfloor O_j \frac{\sum_i c_{ij}}{m} \right\rfloor$ and $\left\lfloor 2 \frac{\sum_i c_{ij}}{n} \right\rfloor$.
- The upper bound M_i is selected between 1 and $n/5$.

algorithm gave better solutions compared to GA and BBA. The GAPs of HGBBA are significantly lower, and the BBA appears as the second-best algorithm by comparing the GAPs. Figure 5, which is a graphic representation of the fitness function using the three discussed algorithms, shows the fast convergence of the BBA and the HGBBA compared to the Genetic Algorithm.

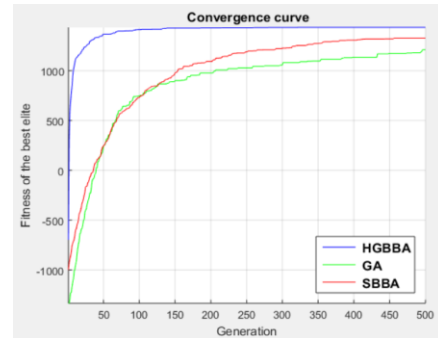


Figure 3 Convergence of fitness function

As reported in Table 1, and for the three categories of the S3 format, HGBBA is significantly better than the algorithms used in the previous works, with a clear difference in performance based on the GAP calculation. However, for the medium and large categories, the HGBBA is considered very competitive and sometimes better than the other algorithms.

In order to test the performance of our proposal, we report the same dataset and conditions from computational experiments of Nobibon et al. [5], and Oliveira et al. [8]. The GAP is calculated when a feasible solution is found by the adopted algorithm. By analyzing Table 1, we clearly deduce that the proposed HGBBA

	S3			M1			M2			L		
	5	10	15	5	10	15	5	10	15	5	10	15
	Nobibon et al. results [5]											
INIT	19.2	17.85	18.34	17.9	16.2	18.36	18.92	19.23	17.84	26.22	24.86	25.65
H8 (TS)	6.86	6.52	7.76	7.22	8.54	7.6	9.75	9.58	9.11	10.86	11.04	10.23
γ	Oliveira et al. results [8]											
0	10.68	6.98	7.77	9.72	9.12	12.15	11.29	9.9	11.15	12.99	12.29	12.15
0.2	6.9	7.14	7.92	7.22	8.67	7.69	10.26	9.59	9.34	11.24	10.94	10.74
0.4	6.79	6.45	7.58	8.47	9.1	7.89	9.68	9.76	9.91	10.86	11.22	10.11
0.6	6.77	6.5	7.49	7.24	8.42	7.63	9.85	9.78	9.44	11.51	11.93	10.22
	HGBBA results											
Init (GA)	23.8	19.05	19.75	20.1	18.8	17.21	19.45	22.88	18.66	27.32	25.67	26.31
HGBBA	6.69	6.51	7.01	7.11	8.55	7.12	9.45	9.77	9.07	10.86	11.01	10.11

Table 1 Experiment results

5 Conclusion

In this paper, we have addressed a very important issue in the marketing strategy of companies in order to optimize their return on investment during advertising campaigns. We have taken up the modeling done by Nobibon et al [5] with the business constraints that must be respected in order to find an optimal solution that maximizes the profit of the company. To test its proposal, Nobibon et al. [5] created a dataset randomly, respecting a certain logic of the upper and lower bounds of each parameter; a job that will produce several dataset categories according to different sizes depending on the number of customers, number of products and the value of Hurdle rate. A dataset on which this paper was based to test our proposal and compare it with the results of Nobibon et al. [5] and the result of other works. Since the problem belongs to the class of NP-hard problems, the use of metaheuristics is required; we have treated the problem based on methods based-population differently to the works cited above and which are all based on neighborhood methods. The Bat Algorithm which is one of the most recent and efficient algorithms was chosen to deal with the problem; it was used in binary version given the discrete nature of the solution space. In order to be able to search in a discrete space, a transformation of the positions of the micro-bats, to discrete values, is essential. The hybridization can provide a more efficient behavior and a higher performance when dealing with large-scale problems. This combination carried out in this paper has shown its effectiveness by remedying the disadvantage of the risk of jamming in a minimum local. As extension of our work, new constraints can be proposed, like reducing products costs where customer respond positively to a big number of products or where the number of customers respond positively to a specific product j . On the other hand, an optimization of HGBBA algorithm can be proposed to reduce the execution time.

References:

- [1] Cohen, M. D., Exploiting response models: optimizing cross-sell and up-sell opportunities in banking, *Information Systems* 29 (2004), pp. 327–341.
- [2] J. McCall, Genetic algorithms for modelling and optimization, *Journal of Computational and Applied Mathematics* 184 (2005), pp. 205–222.
- [3] L. Ryals, Making customer relationship management work: The measurement and the profitable management of customer relationships, *Journal of Marketing* 69 (2005), pp. 252–261.
- [4] M. Y. Smaili and H. Hachimi, "Metaheuristic for optimizing targeted offers in Direct Marketing," 2020 IEEE 6th International Conference on Optimization and Applications (ICOA), Beni Mellal, Morocco, 2020, pp. 1–5, doi: 10.1109/ICOA49421.2020.9094454.
- [5] Nobibon, F. T., R. Leus and F. C. R. Spieksma, Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms, *European Journal of Operational Research* 210 (2011), pp. 670–683.
- [6] R. Faia, T. Pinto, Z. Vale and J. M. Corchado, "Hybrid particle swarm optimization of electricity market participation portfolio," 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, 2017, pp. 1–8, doi: 10.1109/SSCI.2017.8285218.
- [7] Mirjalili, S., Mirjalili, S.M. & Yang, X. Binary bat algorithm. *Neural Comput & Applic* 25, 663–681 (2014).
- [8] T. A. Oliveira, V. N. Coelho, M. J. F. Souza, D. L. T. Boava, F. Boava, I. M. Coelho, B. N. Coelho. A hybrid variable neighborhood search algorithm for targeted offers in direct marketing, *Electronic Notes in Discrete Mathematics* 47 (2015), pp. 205–212.
- [9] Xin She Yang, A new metaheuristic bat-inspired algorithm, in: *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, 2010, pp. 65–74.
- [10] Selim Yilmaz and Ecir U. Kucuksille. Improved Bat Algorithm (IBA) on Continuous Optimization Problems, *Lecture Notes on Software Engineering*, Vol. 1, No. 3, August 2013.
- [11] Vitor N. Coelho, Thays A. Oliveira, Igor M. Coelho, Bruno N. Coelho, Peter J. Fleming, Frederico G. Guimarães, Helena Ramalinho, Marcone J.F. Souza, El-Ghazali Talbi, Thibaut Lust, Generic Pareto local search metaheuristic for optimization of targeted offers in a bi-objective direct marketing campaign, *Computers and Operation Research*, pp 578–587
- [12] Sha-Sha Guo, Jie-Sheng Wang, and Xiao-Xu Ma, Improved Bat Algorithm Based on Multipopulation Strategy of Island Model for Solving Global Function Optimization Problem, *Computational Intelligence and Neuroscience* Volume 2019, Article ID 6068743.
- [13] Lili A. Wulandhari, Siti Komsiyah, Wisnu Wicaksono, Bat Algorithm Implementation on Economic Dispatch Optimization Problem, 3rd International Conference on Computer Science and Computational Intelligence 2018, *Procedia Computer Science* 135 (2018) pp. 275–282.
- [14] Rizk-Allah, R.M., Hassanien, A.E. New binary bat algorithm for solving 0–1 knapsack problem. *Complex Intell. Syst.* 4, pp 31–53 (2018).
- [15] Xingwang Huang,1,2 Xuewen Zeng, and Rui Han, Dynamic Inertia Weight Binary Bat Algorithm with Neighborhood Search, *Computational Intelligence and Neuroscience* Volume 2017, Article ID 3235720.
- [16] Xiao-Xu Ma and Jie-Sheng Wang, Optimized Parameter Settings of Binary Bat Algorithm for Solving Function Optimization Problems, *Journal of Electrical and Computer Engineering* Volume 2018, Article ID 3847951.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US