Revista ARACÊ
ISSN: 2358-2472

COMBINING TABU SEARCH AND GENETIC ALGORITHMS TO SOLVE THE DIRECT MARKETING PROBLEM CONSIDERING CANNIBALISM BETWEEN PRODUCTS

https://doi.org/10.56238/arev6n4-413

Date of submission: 26/11/2024 **Date of publication:** 26/12/2024

Felipe Martins Müller[1], Vanessa Andréia Schneider[2], Iaê Santos Bonilha[3], Veridiane Bárbara de Souza[4], Greici da Rosa da Cruz[5], Michalis Mavrovouniotis[6] and Gédson Mário Borges Dal Forno[7]

ABSTRACT
This paper deals with the problem of the selection of a set of clients that will receive an offer for one or more products during a promotion campaign. Such campaigns are fundamental marketing tools for improving the economic profit of a firm, either by acquiring new customers or by generating additional revenue from existing customers. We work with a well-known mathematical model for the problem. We add the cannibalism constraint to the problem, which avoids some products to be offered simultaneously, to simulate competing products cannibalizing each other's market. We propose a hybrid heuristic, the first by combining a Genetic Algorithm (GA) with Tabu Search (TS). Extensive computational experiments were performed on a set of test problems from literature with and without the cannibalism constraint. We compare our method with a TS and a Matheuristic from literature. The hybrid method outperforms the competing methods in all test cases and all sizes of instances.

Keywords: Direct Marketing Problem. Hybrid Heuristics. Cannibalism.

[1] Federal University of Santa Maria / PPGEP
E-mail: felipe@inf.ufsm.br
[2] Federal University of Santa Maria / PPGEP
E-mail: vanessaandreias@gmail.com
[3] Federal University of Santa Maria / PPGEP
E-mal: iaesb@hotmail.com
[4] Federal University of Santa Maria / PPGEP
E-mail: veridianes@hotmail.com
[5] Federal University of Santa Maria / PPGCC
E-mail: inf.greici@gmail.com
[6] KIOS Research and Innovation Center of Excellence
Department of Electrical and Computer Engineering, University of Cyprus, 2109 Nicosia, Cyprus.
E-mail: mavrovouniotis.michalis@ucy.ac.cy
[7] Federal University of Santa Maria / Department of Applied Computing
Center of Technology, Building 07, 97105-900, Santa Maria-RS, Brazil
E-mail: gedsondf@gmail.com

INTRODUCTION

Recently, Mintel one of the world's leading market intelligence agency released a report proposing seven customer trends that will shape the global markets over the next 10 years. One can summarize these trends as 1) Wellbeing: Seeking physical and mental wellness; 2) Surroundings: Feeling connected to the external environment; 3) Technology: Finding solutions through technology in the physical and digital worlds; 4) Rights: Feeling respected, protected, and supported; 5) Identity: Understanding and expressing oneself in society; 6) Value: Finding tangible, measurable benefits from investments; and 7) Experiences: Seeking and discovering stimulation. (Crabbe et al. 2019). Considering these aspects, one has to think about how to achieve and maintain a customer portfolio by offering good products at the right time. To achieve this feature, several consumer databases have to be exploited. These databases contain robust information about consumer and market, economic, demographic, technological, political, and sociological that helps the analysts to make their decisions.

Promotional campaigns are one of the most direct marketing fundamental tools for client acquisition and overall profit generation (Kotler and Armstrong, 2016), (Abedi, 2017), (Nobibon et al. 2011), (Ładyzynski et al., 2019).

Promotional campaigns typically target clients by considering factors such as the probability of positive response, projected profit, projected cost of the individual offer, and client over-saturation. All these factors generate hard combinatorial problems which are difficult to solve (Praag, 2010), and deserve attention from the academic community (Czikosova, et al., 2014). A very good set of references related to the problem were presented by (Cetin and Alabas-Uslu, 2015), and we will introduce new references here.

Despite the few references of heuristics to solve the direct marketing problem, they tend to be promising approaches as a substitute for statistical methods (Cohen, 2004), (Nobibon et al. 2011). Some examples can be found in (Bhaskar et al., 2009) that proposes the use of fuzzy logic for client selection in a cross-sale marketing campaign from a bank. (Nobibon et al. 2011) proposes a mathematical formulation for the problem as well as two main approaches to solve it: an exact approach and a heuristic approach based on tabu search, which performed very well in small and large instances. They also made available a set of instances and bounds for the problem that we will use of to test and compare the methods proposed in this paper. (Oliveira et al., 2015) presented a hybrid scheme with GRASP and VNS. (Cetin and Alabas-Uslu, 2015) proposed a different approach that divides the problem into two decision problems: assigning products to the market campaign and assigning offers to the client base. The decision problems were solved with linear programming and a heuristic connection between both, achieving good results for all kind of instances. However, since it uses an exact procedure there are limitations when the size of the problem grows and we show here that there are some instances that those procedures are not able to solve.

Recently (Souza, 2018) and (Müller et al. 2019) proposed two hybrid schemes combining GRASP and Genetic Algorithm (GA) with Tabu Search respectively. Then, according to our knowledge, for the first time, the cannibalism constraint was considered in the problem of direct marketing (Souza, 2018), and a new set of test problems was generated and heuristically tested. Cannibalism can be understood as the effect that a product exerts on another making it without attractiveness, i.e., if a customer buys one it will not buy the other. Later on (Schneider, 2019) adapted a mathematical model including some clever constraints to exactly solve almost all instances generated including cannibalism. (Coelho et al., 2017) extended the VNS proposed by (Oliveira et al., 2015), to a bi-objective version of the problem based on the concepts of maximizing profits and searching, at the same time, for a set of customers with less variability over their expected return.

In this paper, we present a Tabu Search with a Genetic Algorithm embedded to deal with cannibalism, named TSeGA. Extensive tests were performed in the same instance set originally proposed by (Nobibon et al. 2011) and we also proposed a new and more difficult set of very large instances and compare TSeGA with two procedures implemented.

In summary, we have made the following contributions: 1) we proposed TSeGA, a new way to hybrid of TS and GA; 2) we included cannibalism constraint and adapt all heuristic and exact procedures to address it; 3) we show that the heuristic methods proposed by (Cetin and Alabas-Uslu, 2015) are not robust because they are not able to solve some of the proposed instances; and 4) we present that TSeGA solves all proposed instances and performs better than the other methods in the literature.

The remainder of this paper is organized as follows. In section 2 we present the problem description, its mathematical model, and a brief description of the other heuristic procedures compared with TSeGA. Section 3 details the TSeGA hybrid heuristic and all procedures related to it. Section 4 presents the computations results and the last section concludes the paper and shows potential ways for future research.

Revista ARACÊ
ISSN: 2358-2472

PROBLEM DESCRIPTION

The direct marketing problem can be split into two decision problems, the first one deciding which products will be participate in the promotional campaign. Recall that promotional campaigns are usually focused on a group of clients and have to be tailored to avoid saturating clients with offers, therefore the choice of which products will be included in the campaign must have a strong impact on the campaign outcomes. The second decision problem is when clients receive offers from the products that are participating in the promotional campaign. These two decisions can be represented by two binary variables: yj = {0 or 1}, which indicates whether the product j is participating in the market campaign or not; and, xij = {0 or 1}, which indicates whether the product j is being offered to the client i or not. For both variables, a value of 1 will indicate the affirmative and a value of 0 will indicate the negative.

In this paper the problem model proposed by (Nobibon et al., 2011) was adapted to cope with cannibalism as showed in (Schneider, 2019). The direct marketing problem is comprised of two main elements: the client set C and the product set P. Each product j ∈ P has a budget; a fixed cost fj which is the fixed cost of the product j participating in the campaign; and an offer quota Oj, which is the minimum number of clients that must receive the offer to make its participation in the marketing campaign justifiable. Each client i ∈ C has a projected profit pij for each offer of a product j; a cost cij associated with each offer of a product j to a client i; a net potential profit NPPij = pij / cij, which represents the return per monetary unit invested in an offer of the product j to the client i; and a limit Mi which is the limit that is placed to simulate offer saturation that could result in a clients' negative response towards the campaign. Then the mathematical model can be defined from equations (1) to (8).

Maximize	$\sum_{i=1}^{m} \sum_{j=1}^{n} (p_{ij} - c_{ij}) x_{ij} - \sum_{j=1}^{n} f_i y_j$	(1)
subject a:	$\sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij} \geq (1+R) [\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{j=1}^{n} f_i y_j]$	(2)
	$\sum_{i=1}^{m} c_{ij} x_{ij} \leq B_j y_j, j = 1, \dots, n,$	(3)
	$\sum_{j=1}^{n} x_{ij} \leq M_{i,} \, i = 1, \dots, m,$	(4)

	$$\sum_{i=1}^{m} x_{ij} \geq O_j y_j, j = 1, \ldots, n,$$	(5)
	$$\sum_{i=1}^{m} x_{ij} \leq m. y_j, j = 1, \ldots, n,$$	(6)
	$y_i + y_j \leq 1 \forall (i,j) \in Can$	(7)
	$y_j, x_{ij} \in \{0,1\} i = 1, \ldots, m, j = 1, \ldots, n.$	(8)

(7)

The main goal of the direct marketing problem is maximizing the net profit generated by all the offers made in the campaign, see equation (1). To be considered viable, the market campaign must achieve hurdle rate *R*, which is the return per monetary point invested in the campaign, see equation (2). Equation (3) represents that the maximum budget per product is respected and the inclusion of the 0-1 variable y_j in the right-hand side of the equation (3) speeds up the CPLEX run. Equation (4) limits the maximum number of offers per client, while equation (5) defines that if a product *j* takes part in the campaign then at least $O_j > 0$ clients receive an offer, and equation (6) specifies that when a product is not part of the campaign, no clients will receive an offer. Equation (7) defines which products are mutually exclusive, i.e., if the pair of product's indexes belong to the set *Can,* it means that they can not be offered to the clients at the same campaign. Finally, the last set of constraints (8) is the integrality constraints. Observe that if the equation (7) is withdrawn the model is the same as (Nobibon *et al*., 2011).

(Nobibon *et al*., 2011) presented seven heuristics and a tabu search algorithm, that outperform methods used by that time. They also provide a set of instances and their objective value for the optimal solutions found so far by a branch-and-price algorithm. The heuristic procedure presented was straightforward and therefore the tabu search was reimplemented by us for sake of comparisons since the source code was not available. Details of those procedures can be found in the original paper.

(Cetin and Alabas-Uslu, 2015) presented two heuristics to solve the problem, in which they determine the products to be included in a campaign using heuristic rules and then distribute these products to the customers optimally. The strategy is carried out in two phases. In phase I, a linear programming model is utilized to predict which products are selected for or removed from the product campaign. Once the products in the campaign are determined, the product targeting problem is reduced to a kind of generalized assignment problem. In phase II, the products selected in phase I are distributed to clients optimally by another optimization model. The two phases are connected via a heuristic rule. Two alternative heuristic rules, derived from the proposed linear programming in phase I, called H-R1 and HR-2, are suggested to predict the products eliminated from the campaign (or equivalently the products involved in the campaign).

These heuristic rules generate two different procedures that outperform the results presented by (Nobibon *et al*., 2011), but since uses commercial solvers to run the associated LP it has a limited capacity in the dimension of the problem to be treated, and some instances are unsolvable by those methods. This last issue is not commented in the original paper but we devise some instances in this paper that show this issue.

(Oliveira et al., 2015) proposed a hybrid heuristic algorithm, GRASP/VNS to solve the problem. Their procedure outperforms the results of (Nobibon *et al*., 2011), but is outperformed by the results of by (Cetin and Alabas-Uslu, 2015). We believe that since their research was published in the same year they worked simultaneously.

In section 3 we present an innovative hybrid approach, named TSeGA, that solve all instances proposed by (Nobibon *et al*., 2011) and a new set of instances presented here and reach results much better than the other heuristics proposed in this paper. The proposed TSeGA is able to outperform the existing heuristics as shown in section 4.

HYBRID HEURISTIC TSEGA

TSeGA is a Tabu Search with a Genetic Algorithm embedded. All procedures developed were presented in detail in this section.

CONSTRUCTIVE ALGORITHMS

To build the initial population of GA were used three different constructive algorithms: two deterministic and one with randomization. The first constructive algorithm is based on the initial solution procedure for the Tabu Search algorithm proposed in (Nobibon *et al*., 2011) with modifications to the product selection procedure; the second is a greedy randomized version of the first, and; the third starts with an infeasible solution applying a repairing procedure to ensure feasibility.

The first constructive algorithm follows the same offer set construction logic proposed in (Nobibon *et al*., 2011), based on $NPP_{ij} = (p_{ij} - c_{ij})/c_{ij}$, with an additional procedure for product inclusion, and the ability to handle cannibalism. With the rule to prioritize higher profit offers, there is a possibility of feasible offer sets to be disregarded, especially when the budget tight. For example, the case when a product has a tight budget and the majority of its most profitable offers have a high cost, then the resultant offer set will reach the maximum budget before reaching the minimum number of offers required for the product. To address this limitation, an additional step is included at the point where the product offer set is completed by the original procedure. For each product not included in the solution, this additional step tries to build a new offer set, based on c_{ij}, instead of NPP_{ij}. If a feasible offer set is found with a higher profit than the original one is then replaced. For the second constructive algorithm, we utilized a greedy randomized version of the first algorithm, that is, instead of picking offers in decreasing order of profitability, it will select a random offer among the 10% best offer available.

The third algorithm is a deterministic algorithm that approaches the solution building process from a different angle, that is, starting with an infeasible solution to convert it into a feasible solution by applying a repairing procedure. This approach starts with an offer set for each product that extrapolates each product budget, while ignoring any problem restrictions. After each offer set is produced, the algorithm tries to repair each product offer set into the feasibility, going in decreasing order of profitability of the infeasible offer sets. We give higher importance to the feasible part of the offer set, counting all profit provided by feasible offers and only 20% of the profit achieved by the infeasible offers. To repair the solution into a feasible one, this process tries to manage offer conflicts caused by more products being offered to a client than its offer limit. This repair will be done by replacing the conflicting offer on one of the products in a way that none other problem restriction will be violated.

The solutions produced by the three constructive algorithms discussed previously are used as initial solutions for the Tabu Search Algorithm. They are also the base of the initial population for the Genetic Algorithm.

TABU SEARCH

The Tabu Search (Glover, 1989) algorithm proposed here drives a local search method with three neighborhoods that are explored sequentially. Neighborhood 1 and 2, say, $N_1(x,y)$ and $N_2(x,y)$, were proposed by (Nobibon et al., 2011) and can be formalized as follows: The set $N_1(x,y)$ contains the feasible solutions (x', y') obtained from (x, y) by considering two clients i and h, and a product j satisfying $y_j = 1$; $x_{ij} = 1$ and $x_{hj} = 0$; then we set $x'_{ij} = 0$ and $x'_{hj} = 1$. The set $N_2(x,y)$ contains the feasible solutions (x', y') obtained from

Revista ARACÊ
ISSN: 2358-2472

(x, y) by considering two clients i and h, and two products j and l satisfying $y_j = y_l = 1$; $x_{ij} = 1$; $x_{hj} = 0$; $x_{il} = 0$ and $x_{hl} = 1$; then we set $x'_{ij} = 0$; $x'_{hj} = 1$; $x'_{il} = 1$ and $x'_{hl} = 0$.

Neighborhood 3, $N_3(x,y)$, described in Algorithm 6, tries to combine the best features of $N_1(x,y)$ and $N_2(x,y)$, and in a way to allow changes in the number of clients allocated to each active product, i.e., the cardinality of the set of clients allocated to each product. At the end it tries to insert new offers to the final solution. The swap movement in $N_3(x,y)$, is represented by a tuple (j, l, i) and means that a client i can be moved from product j to l. Initially, it will explore each client sorted in non-increasing order of their profit variance, trying to find profitable single swaps between active products, and storing up to two most profitable infeasible swaps. Each time such a single swap is found a search for an interchange movement with another client that makes the whole operation feasible is performed. If some interchange movement is found it is immediately done. After all clients were examined there is a final trial to insert profitable offers in the active products. To the best of our knowledge, this is the first neighborhood that effectively changes the number o clients offered by each product, and this feature enables the proposed method to deal with all classes of proposed instances, as shown in the section 4.

The Tabu Search (TS) algorithm (see Algorithm 1) proposed here innovatively traverses the neighborhoods overcoming the convergence rate problems found in (Nobibon *et al*., 2011), mainly in large instances, where each movement causes very low impact in terms of the objective function. The first improvement is to execute any profitable movement as soon as they are found. The size of neighborhoods was also controlled for N_1 and N_2, at the beginning, only 70% of the clients sorted in non-increasing order of their NPP_{ij} are considered for analysis and when no more profitable moves can be found this percentage grows to allow more combinations. Here principles of Variable Neighborhood Search (VNS) were applied (Hansen, P. and Mladenović, 2001). These modifications together with the embedded Genetic Algorithm made the proposed algorithm faster and reaching better solutions than the literature.

Algorithm 1. Tabu Search Procedure

Iter := 0, *tabu_list* := { }, α := 0, choose an initial solution (x, y), $x^* := x' := x$, $y^* := y' := y$
while (*Iter* < 30 and *time* < 300 sec) do
$(x_{prev}, y_{prev}) := (x', y')$
(x_n, y_n) := explore $N_1(x', y')$, considering *tabu_list*
if $of(x_n, y_n) > of(x', y')$, where $of(x, y)$ is the objective function value of the solution (x, y) then
$x' := x_n$, $y' = y_n$, update *tabu_list*

Revista ARACÊ
ISSN: 2358-2472

endif
(x_n, y_n) := explore $N_2(x', y')$, considering *tabu_list*
if $of(x_n, y_n) > of(x', y')$ then
$x' := x_n$, $y' = y_n$, update *tabu_list*
endif
(x_n, y_n) := explore $N_3(x', y')$, considering *tabu_list*
if $of(x_n, y_n) > of(x', y')$ then
$x' := x_n$, $y' = y_n$, update *tabu_list*
endif
if $of(x_{prev}, y_{prev}) <= of(x', y')$ then
Iter := *Iter* + 1
if (*Iter* mod 10) == 0
Update GA Population
Expand TS Neighborhood Size
endif
if *Iter* > 1 then
α := α + 0.03
(x', y') := perform *Genetic Regression* for (x', y') and α
else
(x', y') := perform *Genetic Optimization* for (x', y')
endif
endif
if $of(x^*, y^*) < of(x', y')$ then
$x^* := x'$, $y^* = y'$
Iter := 0, α := 0.0
Reset TS Neighborhood Size
endif
enwhile

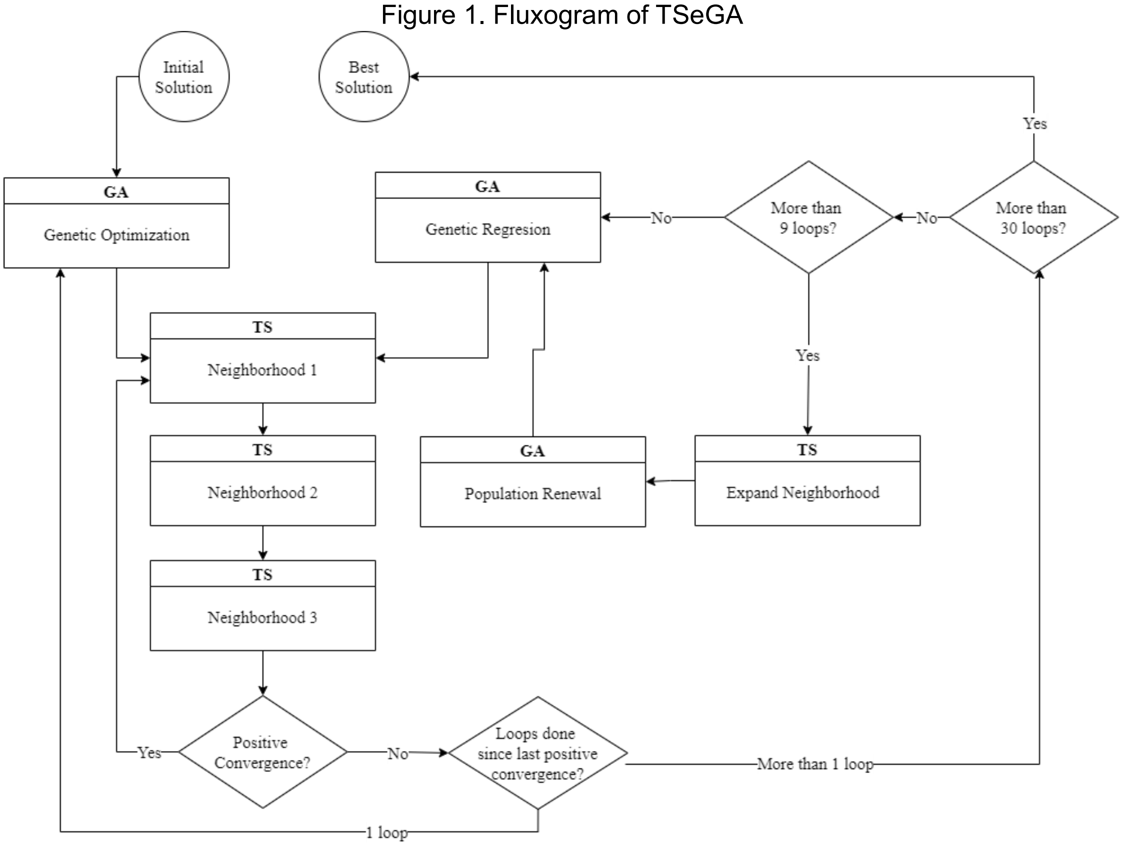
Since TS can perform multiple movements in each iteration the tabu list is adapted in the following way. The movement attribute is the tuple (*j*, *l*, *i*), and meaning that a client *i* can be moved from product *j* to *l*, recall, that for neighborhoods 1 and 2, *j* = *l.* Then anytime that a non-improvement movement is performed the reversal movement is forbidden only for 2 iterations, and that was the best to compromise between solution quality and computational effort. Considering that TS explores exhaustively all neighborhoods in each iteration, and each movement generally causes short increases in the objective function value another innovation was the use of the Genetic Algorithm as a pool of possible alternative ways to continue the search when the TS can not find improved movements for two consecutive iterations. At this point, TS will request the best solution in the current GA population having a fitness score of at least α (starting at 3%) percent from the current objective function value. When a new incumbent solution is found or a new population is generated the α value is reset otherwise is increased by 3%.

Revista ARACÊ
ISSN: 2358-2472

GENETIC ALGORITHM

The GA is detailed in Algorithm 2, where the reproduction process is applied to the population generating a new one that replaces entirely the old population but keeping the best-evaluated individual (Elite) in the next generation. Each new individual is created by the crossover (Algorithm 9) between two selected parents from the previous population. The parents are selected throughout a weighted roulette wheel based on fitness (in that case the objective function value). This implementation of GA utilizes two types of mutation procedures. The first one will only be active in problem instances with cannibalism and will be applied to each generated individual worst than best solution so far, changing the dominant parent by the lower fitness parent. This will change the order of blocking cannibal products, producing different offspring.

Figure 1. Fluxogram of TSeGA



No long term memory was used in TS, but since the population in GA quickly converges to solutions very similar to TS best solution, a diversification procedure was implemented in GA, as the second type of Mutation, which has a 75% chance to be applied for each generated individual that has lower fitness than its parents. When similarity comparisons between individuals from the GA population are made, it is easy to obtain

Revista ARACÊ
ISSN: 2358-2472

which offers are most frequently made. Then, at the end of each generation, a list of offers (maximum size of 0,1*n)* belong to more than 55% of the individuals in the GA population. The Mutation uses this list to forbid these offers to be selected, the choice is made at random and, whereas its number does not goes exceed half the number of elements on the list. The entire interaction process between GA and TS is shown in Figure 1. Details of the chromosome structure, crossover, and mutation can be found (Müller et al. 2018).

Algorithm 2. Genetic Algorithm Procedure

```
Initialize the GA Population performing 200 times the Algorithm 3:  Greedy Randomized Constructive Algorithm
BS := individual with highest fitness on Population
Generation := 0, Offer_List := { }
while (Generation < 4)
    New_Population := { }
    Elite := BS
    while (|New_Population| < 199)
        Parent1 := fitness_weighted_roulette_wheel (Population)
        Parent2 := fitness_weighted_roulette (Population \ {Parent1})
        Dominant := individual with highest fitness between Parent1 and Parent2
        Parent2 := individual with lowest fitness between Parent1 and Parent2
        offspring := apply Algorithm 9 (Crossover) for  Dominant and Parent2
        if (fitness(offspring) < fitness(Elite))
            if {∃ (k,j)/ k ∈ Parent1 and  j ∈ Parent2 and (k,j) ∈ Can}
                Dominant := individual with lowest fitness between Parent1 and Parent2
                Parent2 := individual with highest fitness between Parent1 and Parent2
                offspring := apply Algorithm 9 (Crossover) for Dominant and Parent2
            endif
        endif
        if (fitness(offspring) < fitness(Elite) and Generation > 2)
            if (random (0,1) <= 0.75)
                Apply Mutation to offspring using Offer_List
            endif
        endif
        New_Population :=  New_Population ∪ {offspring}
        if (fitness(offspring) > fitness(Elite))
            Elite :=  offspring
        endif
    endwhile
    Population = New_Population ∪ {Elite}
    if (fitness(Elite) > fitness(BS))
        BS := Elite
    endif
    Offer_List := build the similarity list from New_Population
    Generation := Generation + 1
endwhile
```

Revista ARACÊ
ISSN: 2358-2472

COMPUTATIONAL EXPERIMENTS

We have implemented the two heuristic procedures proposed by (Cetin and Alabas-Uslu, 2015), and TSeGA comparing with the results presented by (Nobibon et al., 2011), and the exact solutions or upper bounds obtained by commercial solvers (GUROBI and CPLEX). The first instances used to test were made available by (Nobibon et al., 2011) named S1, S2, S3, M1, M2 and, L for each combination of m = 100, 200, 300, 1000, 2000, 10000; n = 5, 10, 15; and R = 5%, 10%, 15%; three different random ways to generate the budget B_j and two different ways to generate M_i, comprising 324 instances called Group 1. For each of these instances, we generated a set with (n/5) cannibal pair of products through a statistical analysis identifying the most similar one by Euclidean Distance.

To proof the robustness of TSeGA, we also generate a set of very large instances, the first, named XL, follows the generation method of (Nobibon et al., 2011), all using the intermediate random value of B_j and the tighter M_i with: m = 15000, 20000; n = 10, 20, 30, 40, 50; R = 10% (10 instances); m = 40000; n = 5, 10, 15, 40; R = 10% (4 instances); m = 50000; n = 15; R = 15% (1 instance); and m = 100000; n = 15; R = 15% (1 instance) totaling 16 instances called Group 2. For each of these instances, we generated a set with (n/5) cannibal pair of products through a statistical analysis identifying the most similar one by Euclidean Distance and also by a method called Similarity and Dissimilarity. Similarity and Dissimilarity make use of the best solution found for each instance, then the cannibal pairs are defined as the pair of products offered for the biggest number of identical clients (Similarity) or different clients (Dissimilarity). The results will be presented for each group of instances to better visualize the superiority and robustness of TSeGA. Table 1 sumarizes the computational resources to implement the algorithms.

Table 1. Summary of Implementation Resources

Algorithm	References	Computational Resources
Exact	Schneider, 2019 and Cruz, 2020	IBM ILOG CPLEX and ZIMPL
HR-1	Cetin and Alabas-Uslu, 2015, with heuristic rule 1	Gurobi 8.1.1 with Julia Pro 1.2.0-1 and JuMP
HR-2	Cetin and Alabas-Uslu, 2015, with heuristic rule 2	Gurobi 8.1.1 with Julia Pro 1.2.0-1 and JuMP
TSeGA		Visual Studio Community using C++ (Visual Studio, 2019)

We first define as upper bound (UB) the optimum value of the objective function or the best bound found by the solver and as lower bound (LB) the objective function value

Revista ARACÊ
ISSN: 2358-2472

obtained by the heuristic method. The metric used to compare and evaluate the quality of the methods was the percentual gap defined as Δ = ((UB – LB)/UB).100%, if some instance is not solved by the heuristic method we consider Δ = 100%.

INSTANCES OF GROUP 1

We start the analysis considering 324 instances originally presented by (Nobibon et al., 2011) and also the three kinds of instances including cannibalism. Table 2 shows the frequency of each algorithm in obtaining the best solution. If we look at only Table 2 we can observe that HR-1 has a similar performance to TSeGA, observe Table 3 later on with the percentage gaps, the superiority of TSeGA over HR-1 and HR-2 is highlighted even when we take the average between the best solution found by HR-1 and HR-2.

Table 2. Number of Best Solutions Found by the Algorithms GATeS, HR-1, and HR-2 - Group 1

Heuristic Algorithm	Number of Best Results Found			
	Original Problem	Cannibalism		
		Euclidean Distance (ED)	Similarity	Dissimilarity
TSeGA	108	129	128	100
HR-1	109	75	80	75
HR-2	39	46	55	79
TSeGA = HR-1	3	3	5	4
TSeGA = HR-2	6	5	5	7
TSeGA = HR-1 = HR-2	9	9	10	10
HR-1 = HR-2	50	57	41	49

Table 3 – Average Gaps of TSeGA, HR-1, and HR-2 – Instances of Group 1

Problem	Average GAPs (Δ%)			
	TSeGA	HR-1	HR-2	Best between HR-1 and HR-2
Original Problem	2.09	6.69	4.35	2.12
Cannibalism - ED	1.75	7.20	5.13	2.72
Cannibalism - Similarity	1.73	6.79	5.31	2.95
Cannibalism - Dissimilarity	1.97	7.15	3.71	2.08

Table 4 shows the lack of robustness of HR-1 and HR-2 since they are not able to solve some instances without cannibalism and this behavior intensifies when we include cannibalism.

Revista ARACÊ
ISSN: 2358-2472

Table 4. Number of not Solved Instances by TSeGA, HR-1, and HR-2

Heuristic Algorithm	Number of Not Solved Instances			
	Original Problem	Cannibalism		
		Euclidean Distance (ED)	Similarity	Dissimilarity
TSeGA	0	0	0	0
HR-1	3	1	1	1
HR-2	3	9	9	9

Following we will present the results for Group 2 instances that are bigger and more difficult to solve than Group 1 instances.

INSTANCES OF GROUP 2

The results for Group 2 are summarized in Table 5, where TSeGA shows a performance almost half value of the average gaps from HR-1, HR-2, and the best of both.

Table 5. Average Gaps of TSeGA, HR-1, and HR-2 – Instances of Group 2

Problem	Average GAPs (Δ%)			
	TSeGA	HR-1	HR-2	Best between HR-1 and HR-2
Original Problem	3.47	7.91	6.81	6.59
Cannibalism - ED	4.98	15.41	15.60	14.90
Cannibalism - Similarity	7.30	14.54	19.60	14.52
Cannibalism - Dissimilarity	6.45	13.13	16.83	12.80

As we can see TSeGA performs much better than HR-1 and HR-2 presenting a Gap always better than HR-1 and HR-2 and for instances of Group 1, the Gap is almost half the best Gap between HR-1 and HR-2. We did not present the results from (Nobibon et al., 2011) because HR-1 and HR-2 were always superior, and we also withhold the computational times because the runs were performed in different machines and TSeGA as a time limit of 600 seconds per cycle.

CONCLUSIONS

This paper presented a hybrid heuristic, named TSeGA. It is a Tabu Search with a Genetic Algorithm embedded. The computational results show the quality of TSeGA obtaining better results than other heuristics for large margin when we considering the

Revista ARACÊ
ISSN: 2358-2472

percentage gaps. The inclusion of the cannibalism constraint made the problem more difficult for the other methods while TSeGA could address it without sacrificing its performance.

We also show that HR-1 and HR-2 were not robust since there are instances that they were not able to solve, and this limitation was not discovered before in the literature.

Finally, we can conclude that TSeGA is a promising method to solve the direct marketing problem including or not the cannibalism constraint, and even if it needs some extra computational effort it worthwhile since this problem is a long-term planning.

ACKNOWLEDGMENTS

This work has partially supported by CAPES and by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 739551 (KIOS CoE) and from the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

Revista ARACÊ
ISSN: 2358-2472

REFERENCES

1. Abedi, V. S. (2017). Allocation of advertising budget between multiple channels to support sales in multiple markets. *Journal of the Operational Research Society, 68*(2), 134–146.

2. Bhashar, T., Sundararajan, R., & Krishnan, P. G. (2009). A fuzzy mathematical programming approach for cross-sell optimization in retail banking. *Journal of the Operational Research Society, 60*(5), 717–727.

3. Cetin, F., & Alabas-Uslu, C. (2015). Heuristic solution to the product targeting problem based on mathematical programming. *International Journal of Production Research, 55*(1), 3–17.

4. Coelho, V. N., Oliveira, T. A., Coelho, I. M., Coelho, B. N., Fleming, P. J., Guimarães, F. G., Ramalhinho, H., Souza, M. J. F., Talbi, E., & Lust, T. (2017). Generic Pareto local search metaheuristic for optimization of targeted offers in a bi-objective direct marketing campaign. *Computers and Operation Research, 70*, 578–587. http://dx.doi.org/10.1016/j.cor.2016.09.008

5. Cohen, M. D. (2004). Exploiting response models: optimizing cross-sell and up-sell opportunities in banking. *Information Systems, 29*(4), 327–341.

6. Crabbe, M., Lieberman, G., & Moriarty, S. (2019). Consumer Trends 2030. Disponível em: https://www.mintel.com.

7. Czikosova, A., Antošova, M., & Čukolova, K. (2014). Strategy in direct and interactive marketing communications. *Procedia - Social and Behavioral Sciences, 116*, 1615–1619.

8. Glover, F. (1989). Tabu search: part I. *ORSA Journal on Computing, 1*(3), 190–206.

9. Hansen, P., & Mladenović, N. (2001). Variable neighborhood search: principles and applications. *European Journal of Operational Research, 130*(1), 449–467.

10. Kotler, P., & Armstrong, G. (2016). *Principles of Marketing* (16ª ed.). Pearson, New Jersey.

11. Ładyzynski, P., Zbikowski, K., & Gawrysiak, P. (2019). Direct marketing campaigns in retail banking with the use of deep learning and random forests. *Expert Systems With Applications, 134*, 28–35.

12. Müller, F. M., Bonilha, I. S., & Souza, V. B. (2018). A hybrid approach (genetic algorithm + tabu search) to solve the direct marketing problem. In *Proceedings of XIX CLAIO*, Lima, Peru, 248–255.

13. Nobibon, F. T., Leus, R., & Spieksma, F. C. (2011). Optimization models for targeted offers in direct marketing: exact and heuristic algorithms. *European Journal of Operational Research, 210*(3), 670–683.

Revista ARACÊ
ISSN: 2358-2472

14. Oliveira, T. A., Coelho, V. N., Souza, M. J. F., Boava, D. L. T., Boava, F., Coelho, I. M., & Coelho, B. N. (2015). A hybrid variable neighborhood search algorithm for targeted offers in direct marketing. *Electronic Notes in Discrete Mathematics, 47*, 205–212.

15. Praag, N. V. (2010). Optimization of promotion campaigns using tabu search (Master Dissertation). Faculty of Business and Economics, KULeuven, Belgium.

16. Schneider, V. A. (2019). Análise de modelos matemáticos para o problema de marketing direto com oferta de produtos (Master Dissertation). PPGEP/UFSM, Brazil.

17. Souza, V. B. (2018). Uma abordagem meta-heurística para o problema de marketing direto considerando o canibalismo entre produtos (Master Dissertation). PPGA/UFSM, Brazil.