

Heuristic and Exact Methods to Solve the Direct Marketing Problem Considering Cannibalism between Products

Felipe Martins Müller

Federal University of Santa Maria / PPGE
Center of Technology, Building 07, 97105-900, Santa Maria-RS, Brazil
felipe@inf.ufsm.br

Iaê Santos Bonilha

Federal University of Santa Maria / PPGE
Center of Technology, Building 07, 97105-900, Santa Maria-RS, Brazil
iaesb@hotmail.com

Veridiane Bárbara de Souza

Federal University of Santa Maria / PPGE
Center of Technology, Building 07, 97105-900, Santa Maria-RS, Brazil
veridianes@hotmail.com

Vanessa Andréia Schneider

Federal University of Santa Maria / PPGE
Center of Technology, Building 07, 97105-900, Santa Maria-RS, Brazil
vanessaandreias@gmail.com

Gédson Mário Borges Dal Forno

Federal University of Santa Maria / Department of Applied Computing
Center of Technology, Building 07, 97105-900, Santa Maria-RS, Brazil
gedsondf@gmail.com

ABSTRACT

This paper deals with the problem of the selection of sets 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 adding the cannibalism constraint, that avoids some products to be offered simultaneously. We propose two hybrid heuristics, the first combining GRASP and Tabu Search and the second Genetic Algorithm and Tabu Search. Extensive computational experiments were performed on a set of test problems from literature with and without the cannibalism constraint. For small and medium instances we solved the mathematical model by CPLEX and update the literature bounds, for larger and real life instances the heuristic approaches still seems more suitable.

KEYWORDS. Direct Marketing Problem. Metaheuristics. Exact Methods.

Paper topics. MH – Metaheuristics. OA – Other Applications in OR.

1. Introduction

The technological advances in the last decade generated a substantial increase in the data storage capacity along with a substantial decrease on the costs to store large volumes of data. Aligned with that, the evolution of world wide web provided an environment that facilitates the acquisition and exchange of data allowed companies to build client databases which could offer insights on their preferences and shopping habits. On the other side companies would have more data on their client base that they could process. The need for making sense of such large volumes of information created a great demand for data mining and other decision assisting techniques. Searching competitiveness direct marketing sector apply such techniques to identify which clients would be receptive to a promotional campaign. [Seret et al., 2012].

Promotional campaigns are one of direct marketing's fundamental tools for client acquisition and overall profit generation [Kotler and Armstrong, 2016], [Abedi, 2017], [Nobibon et al. 2011]. The decision of which clients should be target by promotional campaigns, taking into consideration factors such as probability of positive response, projected profit, projected cost of the individual offer, client oversaturation are hard combinatorial problems generally difficult to solve [Praag, 2010].

Despite of the few references about heuristics to solve the direct marketing problem it seems a good approach instead of statistical methods. [COHEN, 2004], [Nobibon et al. 2011]. Some heuristic approaches 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]. propose a mathematical formulation for the problem and solve it exactly and by a 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 make use to test and compare with the methods proposed here. [Oliveira et al., 2015] presented a hybrid scheme with GRASP and VNS. [Cetin and Alabas-Uslu, 2015] proposed an 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 programing with a heuristic connection between both, achieving good results for all kind of instances, but since it uses an exact procedure it presents limitations when the size of the problem grows.

Recently [Souza, 2018] and [Muller et al. 2019] proposed two hybrid schemes combining GRASP and Genetic Algorithm (GA) with Tabu Search respectively. Then, according with 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 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.

In this paper we present the two hybrid approaches Grasp with Tabu Search and GA with Tabu Search adapted to deal with cannibalism. We present the mathematical model modified to included the cannibalism constraint and some changes that improves the CPLEX performance. Extensive tests were performed in the same instance set and we compare with other heuristics and benchmarks from literature. The results and instances are made available to the research community through Research Gate (<https://www.researchgate.net/>).

2. Problem description

The direct marketing problem can be split into two decision problems, the first one being deciding which products will be participating on the promotional campaign. Promotional campaigns are usually focused on a group of clients and have to be tailored to avoid saturating clients with offers so the choice of which products will participate of the campaign has a strong impact on the campaign outcomes. The second decision problem is which clients will receive offers from which products among the products that are participating on the promotional

campaign. These two decisions can be represented by two binary variables: $y_j = \{0 \text{ or } 1\}$, which indicates whether the product j is participating on the market campaign or not; and, $x_{ij} = \{0 \text{ or } 1\}$, 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.

There are several business restrictions that can affect the planning of a market campaign, in this paper the problem model that was 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 $j \in P$ has a budget; a fixed cost f_j which is the fixed cost of the product j participating of the campaign; and an offer quota O_j , which is the minimum number of clients that must receive its offer in order to make its participation in the marketing campaign justifiable. Each client $i \in C$ as a projected profit p_{ij} for each offer of a product j ; a cost c_{ij} associated with each offer of a product j to a client i ; a net potential profit $NPP_{ij} = p_{ij} / c_{ij}$, which represents the return per monetary unit invested in an offer of the product j to the client i ; and a limit M_i which is the limit that is placed in order 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).

$$\text{maximize } \sum_{i=1}^m \sum_{j=1}^n (p_{ij} - c_{ij}) x_{ij} - \sum_{j=1}^n f_j y_j \quad (1)$$

subject to

$$\sum_{i=1}^m \sum_{j=1}^n p_{ij} x_{ij} \geq (1+R) \left[\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{j=1}^n f_j y_j \right] \quad (2)$$

$$\sum_{i=1}^m c_{ij} x_{ij} \leq B_j y_j, \quad j = 1, \dots, n \quad (3)$$

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

$$\sum_{i=1}^m x_{ij} \geq O_j y_j, \quad j = 1, \dots, n \quad (5)$$

$$\sum_{i=1}^m x_{ij} \leq m y_j, \quad j = 1, \dots, n \quad (6)$$

$$y_i + y_j \leq 1, \forall (i, j) \in \text{Can} \quad (7)$$

$$y_i, x_{ij} \in \{0, 1\}, i = 1, \dots, m, j = 1, \dots, n \quad (8)$$

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). In order to be consider viable, the market campaign must achieve hurdle rate R , which is the return per monetary point invested in the campaign, see equation (2). The 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 the 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. The equation (7) defines which products are mutually excludent, 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.

3. Hybrid Heuristics

3.1. GRASP with Tabu Search

The idea here is adapt the Constructive Procedure proposed by [Nobibon et al. 2011] as a Greedy Randomized Adaptive Procedure [Feo and Resende, 1995] and use the obtained solution

as input for a Tabu Search Procedure. All the methods showed here includes the cannibalism constraint, but all of them works when the set $Can = \{ \}$. The randomized constructive procedure was summarized in the Figure 1, where a list of clients is built based on its NPP_{ij} , and the choice is made among the 10% better values. The constructive algorithm is divided into three phases, the first, lines 2-13, comprises the product selection and includes the greedy randomized selection of offers at line 7. The product selected is the one that present the best net profit among all tested. The second phase is the budget maximization from lines 14 to 22 and the last one added to fully prepare this solution for the subsequent tabu search procedure, when we fill it up the products with offers that do not leads to some infeasibility, see lines 23 to 27.

Figure 1. Constructive Greedy Randomized Adaptive Procedure

```

1: total_profit = 0; total_cost = 0;  $V \leftarrow \text{client\_list}$ ;  $Sol = \{ \}$ ; random_offers =  $\{ \}$ ; number_offers = 0
2: for each product  $j$  and for each client in  $V$ , compute  $NPP_{ij} = p_{ij} / c_{ij}$ 
3: sort the clients in  $V$  in non-increasing order of their  $NPP_{ij}$ 
4: for each product  $j \notin Sol$  and not cannibal with any other product belonging to  $Sol$  do
5:   random_offers =  $\{ \}$ 
6:   while number_offers <  $O_j$ 
7:     the offer of the product  $j$  to the client  $i$  is chosen randomly among the 10% clients with
       higher values of  $NPP_{ij}$  and inserted in random_offers
8:     number_offers = number_offers + 1
9:   endwhile
10:   $C_j$  = cost of the offers in random_offers
11:   $P_j$  = profit of the offers in random_offers
12:   $PR_j = P_j - C_j - f_j$ 
13: endfor
14: select the product  $j^*$  with the highest  $PR_j$  satisfying  $C_{j^*} \leq B_{j^*}$  and
     $total\_profit + P_{j^*} \geq (1+R)(total\_cost + C_{j^*} + f_{j^*})$ 
15: if  $j^*$  exists and  $PR_{j^*} > 0$  then
16:    $Sol = Sol + \{j^*\}$ ;  $y_{j^*} = 1$ ; total_profit = total_profit +  $P_{j^*}$ ; total_cost = total_cost +  $C_{j^*}$ 
17:   for each client  $i$  amongst the first  $O_{j^*}$  clients in the sorted list do
18:      $x_{ij^*} = 1$ ;  $M_i = M_i - 1$ 
19:     if  $M_i = 0$  then remove client  $i$  from  $V$  endif
20:   endfor
21:   goto 4
22: endif
23: for each product  $j \in Sol$  do
24:   for each client  $i \in V$  do
25:     if  $p_{ij} > c_{ij}$  and the offer of product  $j$  to client  $i$  leads to a feasible solution,
       make that offer and update the current solution
26:   endfor
27: endfor

```

The constructive procedure showed in Figure 1 acts like an initial solution to a Tabu Search procedure [Glover, 1989], and it performs five loops keeping the best solution at the end.

For the tabu search algorithm implementation, we implemented the procedure proposed by [Nobibon et al., 2011] with two modifications. We eliminate the diversification procedure, because we conclude that it was ineffective for the most of the situations with a very high computational effort.

The second change made was the inclusion of a fourth neighborhood which consists in the permutation of a single offer from a product to another that is not being offered to the client receiving the offer in question. The main reason for adding this neighborhood is the fact that none of the three neighborhoods proposed by [Nobibon et al., 2011] are capable of changing the solutions cardinality (client numbers per product) which was a limiting for the search procedure. Neighborhood1 makes a single swap, i.e., for all $j = 1$ and $i_1 \neq i_2$, if $(x_{i_1j} = 1$ and $x_{i_2j} = 0)$ make $(x_{i_1j} = 0$ and $x_{i_2j} = 1)$. Neighborhood2 makes a double swap, i.e., for all $j_1 = 1$, $j_2 = 1$, $j_1 \neq j_2$ and $i_1 \neq$

i_2 , if $(x_{i_1j_1} = 1, x_{i_1j_2} = 0, x_{i_2j_1} = 0 \text{ and } x_{i_2j_2} = 1)$ make $(x_{i_1j_1} = 0, x_{i_1j_2} = 1, x_{i_2j_1} = 1 \text{ and } x_{i_2j_2} = 0)$. Neighborhood3 tests all possible feasible swaps between all pair of clients in all active products. Neighborhood4 makes a permutation, i.e., for all $j_1 = 1, j_2 = 1, j_1 \neq j_2$ and $i_1 \neq i_2$, if $(x_{i_1j_1} = 1 \text{ and } x_{i_2j_2} = 0)$ make $(x_{i_1j_1} = 0 \text{ and } x_{i_2j_2} = 1)$. Tabu Search Procedure can be see in Figure 2.

Figure 2. Tabu Search Procedure

```

1: Current_Sol = Current_Best_Sol; Iterations = 0
2: while Iterations < 50 and Time < 1 hour
3:   Sol_Neighborhood1 = neighborhood_1(Current_Sol)
4:   if (Sol_Neighborhood1 > Current_Best_Sol) then
5:     Current_Sol = Sol_Neighborhood1
6:     Current_Best_Sol = Sol_Neighborhood1
7:   else
8:     Sol_Neighborhood2 = neighborhood_2(Current_Sol)
9:     if Sol_Neighborhood2 > Current_Best_Sol then
10:      Current_Sol = Sol_Neighborhood2
11:      Current_Best_Sol = Sol_Neighborhood2
12:    else
13:      Sol_Neighborhood3 = neighborhood_3(Current_Sol)
14:      if Sol_Neighborhood3 > Current_Best_Sol then
15:        Current_Sol = Sol_Neighborhood3
16:        Current_Best_Sol = Sol_Neighborhood3
17:      else
18:        Sol_Neighborhood4 = neighborhood_4(Current_Sol)
19:        if Sol_Neighborhood4 > Current_Best_Sol then
20:          Current_Sol = Sol_Neighborhood4
21:          Current_Best_Sol = Sol_Neighborhood4
22:        else
23:          Current_Sol = best solution not tabu between all neighborhoods
24:        endif
25:      endif
26:    endif
27:  endif
28:  update Tabulist
29:  Iterations = Iterations + 1
30: endwhile

```

3.2. Genetic Algorithm with Tabu Search

Noting that the results obtained by the Constructive Greedy Randomized Adaptive Procedure presented in Figure 1 were still with a very poor quality as an initial solution (ranging from 25% to 60% far from the instances best known upper bounds). We decide to try a Genetic Algorithm (GA) [Holland, 1992] as an intermediate step in order improve the local search performance. The procedure starts with a greedy randomized constructive procedure in order to build a GA initial population. After a fixed number of generations the best solution found by GA is used as Tabu Search (TS) starting point. The GA with TS can be summarized in Figure 3.

Figure 3. Genetic Algorithm with Tabu Search Procedure

```

1: Population = { }; Sol = { }; Result = { }; Final_Result = { };
2: while Population_size < 100
3:   Sol = Constructive_Greedy_Randomized_Adaptive_Procedure()
4:   Population = Population + {Sol}
5: endwhile
6: Result = Genetic_Algorithm_Procedure(Population)
7: Final_Result = Tabu_Search_Procedure(Result)

```

As we can see in Figure 2, the Constructive Greedy Randomized Adaptive Procedure and the Tabu Search Procedure are the same presented in section 3.1, then the Genetic Algorithm Procedure will be presented in Figure 4 and detailed as follows.

Figure 4. Genetic Algorithm Procedure

```

1: Population = {generated by Constructive_Greedy_Randomized_Adaptive_Procedure( )};
2: Best_Solution = evaluate_best_solution (Population)
3: while Generation_Number < 30
4:   New_Population = { }
5:   while Population_Size < 99
6:     Parent_Solution1 = fitness_weighted_roulette(Population)
7:     Parent_Solution2 = fitness_weighted_roulette(Population - Parent_Solution1)
8:     New_Solution = crossover( Parent_Solution1, Parent_Solution2)
9:     New_Population = New_Population + {New_Solution}
10:    if New_Solution > Elite_Solution
11:      Elite_Solution = New_Solution
12:    endif
13:  endwhile
14:  Population = New_Population + {Best_Solution}
15:  if Elite_Solution > Best_Solution
16:    Best_Solution = Elite_Solution
17:  endif
18: endwhile
19: return Best_Solution

```

The main objective of the GA was to be a high coarse convergence and fast algorithm that would produce better initial solutions for the TS. As such, it was decided that the GA should not be focused on individual offer, focusing itself in small correlated groups of possible offers. With that objective in mind, the chromosome for the GA was built focused on clients, each client being represented by a gene in the chromosome. The genes have a list of every possible offer to be made to the client it represents (one from each product). Each gene also has values for revenue, cost, hurdle rate and available offer slots for the particular clients, which are used to accelerate the crossover operations. The chromosome itself was built as a linked list of genes (clients) along with some preprocessed information which is: its fitness(total profit); total revenue; total cost; remaining budget for each product; and a boolean vector indicating which products are being offered in the solution. The structure of the chromosome can be observed in Figure 5 and entire crossover procedure can be seen in Figure 6.

Figure 5. Chromosome Structure

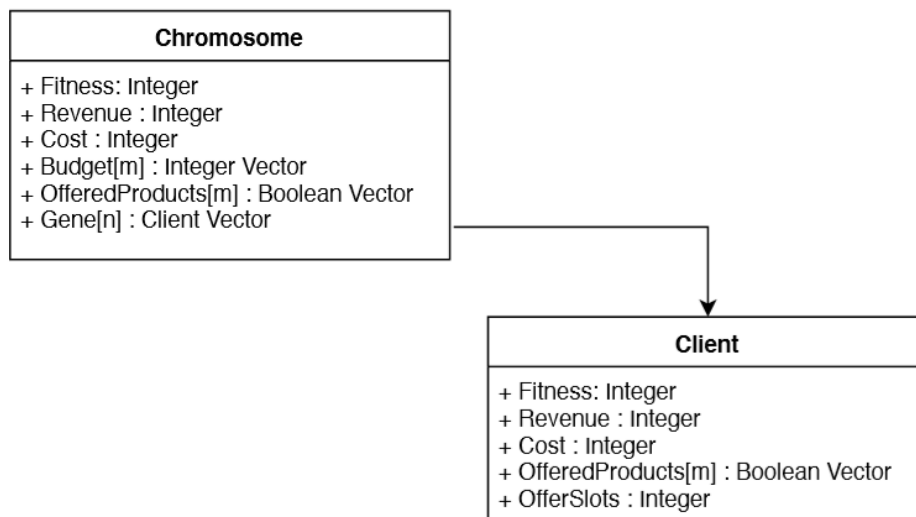


Figure 6. Crossover Procedure

```

1: New_Solution = { }; Losing_Genes = { };
2: if there is one or more  $(j_1, j_2) \in Can$ ,  $j_1 \neq j_2$ ,  $j_1$  in Parent_Solution1 and  $j_2$  in Parent_Solution2
3:   set the products in Can from Parent_Solution with lower objective function value as inviable
4: endif
5: for  $i = 1$  to Gene_Quantity
6:   if Parent_Solution1.gene[i] is more profitable than Parent_Solution1.gene[i]
7:     then
8:       if a product  $j$  in Parent_Solution1.gene[i] is not able to meet its offer quota
9:       then eliminate all previous offers of product  $j$  from the solution and mark it as inviable
10:      transfer all feasible offers in Parent_Solution1.gene[i] to New_Solution.gene[i]
11:      insert Parent_Solution2.gene[i] into Losing_Genes
12:    else
13:      if a product  $j$  in Parent_Solution2.gene[i] is not able to meet its offer quota
14:      then eliminate all previous offers of product  $j$  from the solution and mark it as inviable
15:      transfer all feasible offers in Parent_Solution2.gene[i] to New_Solution.gene[i]
16:      insert Parent_Solution1.gene[i] into Losing_Genes
17:    endif
18:  endfor
19: for each product  $j$  in New_Solution that has not meet its offer quota do
20:   insert offer of product  $j$  in New_Solution to the clients in non decreasing order of  $NPP_{ij}$  until the
   quota is achieved
21: endfor
22: for each gene  $g$  belonging to Losing_Genes and each product  $j$  in New_Solution do
23:   transfer all feasible offers from  $g$  to New_Solution
24: endfor
25: for each product  $j$  in New_Solution that is under its budget do
26:   insert offer of product  $j$  in New_Solution to the clients in non decreasing order of  $NPP_{ij}$  until
   there are no more budget available
27: endfor
28: return New_Solution

```

The crossover was implemented in order to transfer the best client offer configuration among the parent solutions to the new solution. In other words, each individual client status (set of offers) from one parent is compared with the same client from its counterpart, the best client status (most lucrative) would be transferred to the new solution. It is important to point out that this initial idea still does not guarantee solution integrity and new support procedures need to be developed as explained. This algorithm was initially implemented to solve direct marketing problem without cannibalism. In order to adapt the algorithm to deal with problem instances containing cannibalism between products, before the first step of the crossover a cannibalism check is performed (Figure 6, lines 2-4). If it is detected that there are products in one parent solution that are cannibal pairs of products offered on the second parent solution, the cannibal products from the inferior parent solution (solution with lower fitness) will be left out of the crossover procedure.

The offer checking procedure (Figure 6, lines 5-18) is able to guarantee that infeasible offers will not be made but this still does not guarantee that the final solution will be feasible due to two factors. The first factor is that parents can have different product configurations which will most likely lead to a solution where products will not be able to achieve their minimum offer requirement. The second factor is that even if both parents have the same product configuration, it is still not guaranteed that all the products solution will achieve their offer quota due to the fact that the final offer distribution can be unbalanced.

After all genes have been evaluated and the offers transferred to the new solution, the next step is to guarantee that every offered product meet its offer quota (Figure 6, lines 19-21). Thus, the most profitable available offers will be included for each product that was offered but did not meet its offer quota, until these products meet their quota. If during this procedure, a

product still fails to meet its quota, that product is eliminated from the solution in order to keep the solution feasible.

After the previously mentioned process, the algorithm has a feasible initial solution but, since the objective of this algorithm is to provide a good quality solution, the remaining budget on the offered products must be exhausted. This procedure was divided into two parts: the genetic transfer process and the new offers addition process. On the genetic transfer process, each gene that was not chosen during the initial crossover will be verified for additional feasible offers (Figure 6, lines 22-24). After the genetic transfer process, each product that still has budget available will have additional feasible offers added according to the ordered client list (Figure 6, lines 25-27).

During tests done on the earlier stages of implementation, the order in which the genes are evaluated also proved highly impact on GA. Since the GA will compare parents genes in a predetermined order, it is natural that the earlier genes would have a higher impact on the solution than the later genes. The order of the gene list was based on profit prospect of the clients, that value being the sum of all profitable offer prospects (disregarding any nonlucrative offers) for each client.

The reproduction procedure for the GA chooses the parents by a weighted roulette. Two parents are picked generating a single new solution which will be transferred to a new population, this continues until the new population is completely filled. Once the new population is filled, the elite solution from the old population will replace the worst solution from the new population, and then the new population will replace the old one. Due to the constructive nature of the algorithm it was opted for no mutation procedures.

The initial population for the GA was created by the Constructive Greedy Randomized Adaptive Procedure presented in Figure 2. The stopping criteria for the GA is 30 generations, once it is reached, the elite solution from the final population will be improved by the Tabu search algorithm.

4. Computational Experiments

The set of instances was the one originally proposed by [Nobibon et al., 2011] and it was used for sake of comparisons of all methods developed for this problem. The instance set consists of 324 problem instances divided into 3 categories: small instances, medium instances and large instances. The small instances category is divided into 3 sub-categories: S1 (100 clients), S2 (200 clients) and S3 (300 clients). The medium category is divided into 2 sub-categories: M1 (1000 clients) and M2 (2000 clients). Lastly the large (L) category contains instances with 10000 clients. For each of these categories there were three instances for each combination of the following attributes: hurdle rates (5%, 10% or 15%), number of available products (5, 10 or 15) and gap between clients profitability (s for small and l for large).

The set *Can* that contains the pairs of cannibal products is built using cluster analysis in the original set of instances to identify similarities that can give a hint on each pair of products will not be offer simultaneously. The metric adopted was the minimum Euclidian distance between each client profit expectance. The closest ones were placed in set *Can*, considering the following: 1 pair of cannibal products for instances with 5 products, 2 pairs of cannibal products for instances with 10 products and 3 pairs of cannibal products for instances with 15 products.

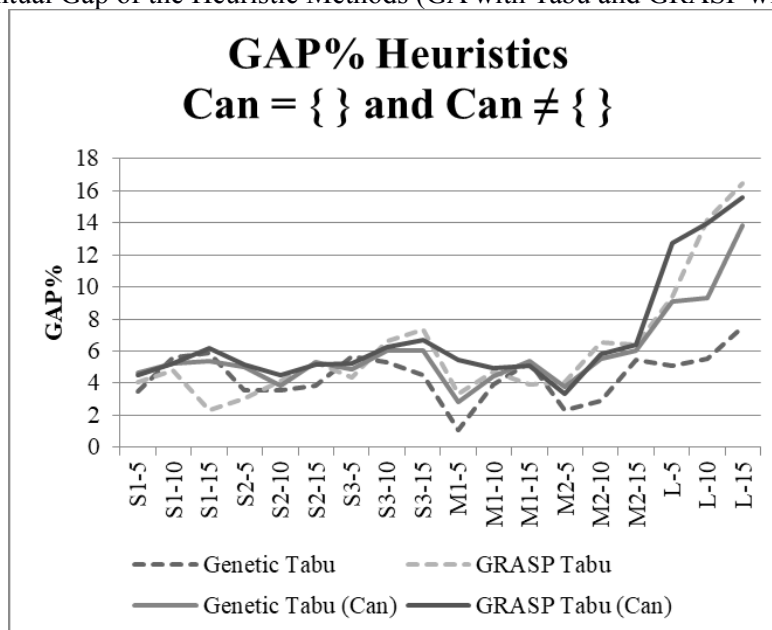
All heuristic algorithms were implemented in C. In order to make a fair comparison between them, we left the algorithms runs from one hour or until it finds the optimal solution. For the GRASP with Tabu Search algorithm, 5 random resets were utilized with solutions generated by the Constructive Greedy Randomized Adaptive Procedure (Figure 2). For the Tabu Search algorithm, we have set the tabu list size to 20 and the maximum iterations before stopping to 30. For the GA with Tabu Search the population size was set to 100. The GA was set to run for 30 generations before the elite solution is sent to be processed by the Tabu Search which parameters where kept the same.

Considering that the cannibalism relationship modifies the behaviour of the original problem, there was a high probability that the optimal value for some instances would change, and since there was no other mention of cannibalism in the literature we decided to use the IBM ILOG CPLEX Optimization Studio to have a performance measure for this problem. From the 324 instances, the solver was able to achieve the optimum on 298 instances (with a time limit of 1 hour) which corresponds to 91,98% of the instances. Among these results, 140 instances presented the same optimum as the original problem (with $Can = \{ \}$), corresponding to 43,21% was not the most appropriate. Solving the problem exactly, with $Can \neq \{ \}$, seems to be easier than when $Can = \{ \}$, because how bigger is Can more restricted is the set of solutions and the cuts generated to find the integer solutions works better in the CPLEX.

In order to measure the algorithms performance, we use the following formula: $GAP\% = ((Z_{UB} - Z_{HA})/Z_{UB}) * 100$, that is the percentage difference between the solution obtained by the heuristic algorithm (Z_{HA}) and the optimal solution or the upper bound (Z_{UB}) obtained by the solver. The results from the algorithm presented in this paper from $Can = \{ \}$ and $Can \neq \{ \}$, can be seen in Figure 7.

From the Figure 7 we can observe that the heuristic performance is equivalent and the GAP is growing with the size of the problem. This fact is due to two problem, first is that the problem is NP-complete and become more difficult as its size grows, but at the same time the quality of the Upper Bounds is degenerating, justifying even more the use of heuristics for larger instances.

Figure7. Percentual Gap of the Heuristic Methods (GA with Tabu and GRASP with Tabu)



To have a better idea of the behavior of the proposed heuristic and the other methods found in the literature we condensed the results in Figure 8. In the leftmost column we can found the results for $Can = \{ \}$ and $Can \neq \{ \}$. All the intersections lines x columns filled with numbers are the average value for all instances from the considered subset. The intersections left empty means that there were no results in the literature.

The lines labeled Objective Value are the average value of the optimum value solution or the upper bound obtained by the solver, since we run again all the instances we update the bounds for $Can = \{ \}$ and we present the bounds for $Can \neq \{ \}$. The lines presenting the heuristic results show the average percentual GAP for each subset. We will focus our analysis on the case on $Can \neq \{ \}$, because the other case was extensively studied and we just bring the results for sake of comparisons.

Figure 8. Results from Exact and Heuristic Methods

Algorithm	S1			S2			S3			MI			M2			L		
	5	10	15	5	10	15	5	10	15	5	10	15	5	10	15	5	10	15
Can = {}	Objective value			816	1643	2435	1628	3269	4554	4703	7018	8346	15751	22710	16547	31390	48767	85414
	Genetic Tabu			3.47	5.60	5.90	3.56	3.57	3.84	5.68	5.31	4.50	1.05	3.90	5.34	2.34	2.88	5.48
	GRASP Tabu			4.04	4.82	2.32	3.06	4.12	5.28	4.36	6.61	7.32	3.34	4.74	3.94	4.09	6.53	9.36
Can ≠ {}	Nobibon et al. [2011]									6.86	6.52	7.76	7.22	8.54	7.60	9.75	9.58	10.86
	Oliveira et al. [2014]									6.77	6.45	7.58	7.22	8.42	7.63	9.68	9.59	10.86
	Cetin & Alabas-Uslu [2015]												5.04	0.86	2.28	0.00	0.91	1.24
Can ≠ {}	Objective value			793	1599	2378	1617	3189	4359	2411	4587	6772	8258	14960	21966	16227	30842	46579
	Genetic Tabu			4.65	5.23	5.35	5.03	3.82	5.29	4.86	6.05	6.00	2.80	4.42	5.39	3.75	5.50	6.06
	GRASP Tabu			4.49	5.24	6.15	5.14	4.52	5.13	5.27	6.23	6.67	5.47	4.96	5.11	3.34	5.80	12.76

From Figure 8 we can see that GA with Tabu is consistently better than GRASP with Tabu mainly in larger instances when this difference became larger. The percentual gaps are not so different from the case with $Can = \{ \}$ which shows that the adaptation of the heuristic procedures was successful and produced very good results.

Since the problem proposed is new all the results, instances and bounds will be made available for the scientific community through Research Gate (<https://www.researchgate.net/>). We imagine that this can improve the research and the competitiveness among the methods developed for this kind of problems.

5. Conclusions

We presented a new version of the direct marketing problem including the cannibalism restrictions. This problem for our knowledge was never mentioned in the previous literature being an open field of research.

Two hybrid heuristic procedures were developed and presented very consistent results, GA with Tabu Search have a small advantage over GRASP with Tabu Search and must be used for real life instances that have higher dimensions that we can solve exactly as showed here.

A mathematical model was presented and solved by CPLEX giving us some optimal solutions and upper bounds that allow us to show the quality of the heuristic methods developed here.

During our tests the solver was able to obtain optimal solutions for most problems on small and medium categories. The solver started showing some difficulties on achieving optimal solutions for larger problems (L instances), still being able to find the optimal solution for the majority. Even though the exact method was able to find the optimal solution for majority of the problem instances, these difficulties shown on higher problem sizes suggest that, on problems with realistic sizes (hundreds of thousands of clients), these methods may become ineffective. In this scenario, methods such as the heuristic proposed in this paper may produce better solutions due to their less computational intensive nature.

The algorithm proposed in this paper is a work in progress and up until this point we limited our tests to the instance set proposed by (Nobibon et al., 2011) due to the fact that is an well known instance set, giving us basis for comparison with other approaches in the literature. For the future we intend to construct an instance set with larger problems as well as give more reality to the metric used to simulate cannibalism.

References

- 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.
- Bhashar, T., Sundararajan, R. and Krishnan, P. G. (2009). A fuzzy mathematical programming approach for cross-sell optimization in retail banking. *J. Oper. Res. Society*. 60(5):717-727.
- Cetin, F. and Alabas-Uslu, C. (2015). Heuristic solution to the product targeting problem based on mathematical programming. *Int. Journal of Production Research*. 55(1):3-17.
- Cohen, M. D. (2004). Exploiting response models: optimizing cross-sell and up-sell opportunities in banking. *Information Systems*. 29(4):327-341.
- Feo, T. and Resende, M. A. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*. 6:1-27.
- Glover, F. (1989). Tabu search-part I. *ORSA Journal on Computing*. 1(3):190-206.

Holland, J.H. (1992). Genetic Algorithms. *Scientific American*. July:66-72.

Kotler, P. and Armstrong, G. (2016). Principles of Marketing. 16 ed., Pearson, New Jersey.

Muller, F. M., Bonilha, I. S. and Souza, V. B. (2018). A Hybrid Approach (Genetic Algorithm + Tabu Search) to Solve the Direct Marketing Problem. In: Proceedings of XIX CLAIO. Lima - Peru.

Nobibon, F. T., Leus, R. and Spieksma, F. C. (2011). Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms. *EJOR*, 210(3):670-683.

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

Praag, N. V. (2010). Optimization of Promotion Campaigns Using Tabu Search. Master's Thesis. Faculty of Business and Economics, KULeuven, Belgium.

Seret, A., Verbraken, T., Versailles, S. and Baesens, B. (2012). A new SOM-based method for profile generation: Theory and an application in direct marketing. *EJOR*, 220(1):199-209.

Schneider, V. A. (2019) Análise de Modelos Matemáticos para o Problema de Marketing Direto com Oferta de Produtos. Dissertação de Mestrado, PPGE/UFSM, Brazil.

Souza, V. B. (2018). Uma Abordagem Meta-heurística para o Problema de Marketing Direto Considerando o Canibalismo entre Produtos. Dissertação de Mestrado, PPGA/UFSM, Brazil.