A hybrid variable neighborhood search algorithm for targeted offers in direct marketing

T. A. Oliveira [a,1,2] V. N. Coelho [b,1,3] M. J. F. Souza [a,1,4]
D. L. T. Boava [a,1,5] F. Boava [a,1,6] I. M. Coelho [c,1,7]
B. N. Coelho [a,1,8]

[a] *Federal University of Ouro Preto, Ouro Preto, MG, 35400-000, Brazil*

[b] *Federal University of Minas Gerais, Belo Horizonte, MG, 31270-901, Brazil*

[c] *Fluminense Federal University, Niterói, RJ, 24210-240, Brazil*

Abstract

This paper focuses on the targeted offers problem in direct marketing campaigns. The main objective is to maximize the feedback of customers purchases, offering products for the set of customers with the highest probability of positively accepting the offer and, at the same time, minimizing the operational costs of the campaign. Given the combinatorial nature of the problem and the large volume of data, involving instances with up to one million customers, approaches solely based on mathematical programming methods, said exact, appear limited and infeasible. In this paper, the use of a hybrid heuristic algorithm, based on the Greedy Randomized Adaptive Search Procedures and General Variable Neighborhood Search, is proposed. Computational experiments performed on a set of test problems from the literature show that the proposed algorithm was able to produce competitive solutions.

Keywords: Direct marketing, Targeted offers problem, GRASP, General Variable Neighborhood Search, Operational Research

1   Introduction

This work deals with the Targeted Offers problem in Direct Marketing Campaigns – TODMC. The main goal of this problem is to determine which customers should receive the offer of several products, in order words, find the set of potential customers to receive certain offers, maximizing profits and respecting operational requirements.

Direct Marketing – DM campaigns produces, along time, important data and information for decision support, which should be carefully analyzed. Nowadays, with the rise of the analyses dealing with large volume of data, also known as big-data analysis, this task have been showing up ever more vast and complex [9]. In this sense, the need of integration between DM and Operations Research - OR is eminent, which has been presenting problems more and more complex and intractable without the appropriate tool. Researchers have been developing efficient and intelligent models since the 80's [6]. In this context, OR has been playing a big role in satisfying and attending costumers, what can be verified in recent literature works relating both areas [3,1,10,2].

Cohen et al. [3] present a case study of TODMC applied to a campaign involving a large international bank, 11 types of offers and 5 different investments were considered. Approximately 2,500,000 million potential customers were available in that campaign. From response models, the benefit of each offer for each customer was estimated. The results and analysis had validated the proposal, which was promising to increase the success of future campaigns. Bhaskar et al. [1] proposed a fuzzy model, including some uncertainty over the final results. This approach proved to be applicable and efficient, given the imprecision in predicting consumer reactions.

The model presented by Nobibon et al. [10], state-of-art, embraces similar constraints as Cohen et al. [3], but also, incorporating new restrictions, such as the minimum number of units of each product offered during the campaign, known as Minimum-Quantity Commitment – (MQC). Eight different

[2] Email: thaysoliveira7@gmail.com
[3] Email: vncoelho@gmail.com
[4] Email: marcone@iceb.ufop.br
[5] Email: profboava@yahoo.com.br
[6] Email: profamacedo@yahoo.com.br
[7] Email: imcoelho@ic.uff.br
[8] Email: brunonazario@gmail.com

algorithms were proposed by these authors, seven of them were based on exact approaches and the other were based on the Tabu Search – TS heuristic [5]. Among these proposed methods, the Branch-and-Price and TS showed the best performance.

The TODMC model presented by Nobibon et al. [10] can be rewritten as follows: Given a set of clients $C = [c_1, c_2, ..., c_m]$ and a set of offers $O = [o_1, o_2, ..., o_n]$, a cost $d_{ij} > 0$ and profit $r_{ij} \geq 0$ is associated to each offer $j \in O$ targeted to a given customer $i \in C$. For each client $i \in C$, there is a number of maximum offers M_i, which indicates the maximum of products that could be offered to that client. For each product offer $j \in O$, there is a minimum number of products O_j^{min} that should be offered during the campaign, minimum expected profit R (also known as hurdle rate), available budget B_j for each and, finally, a fixed cost f_j is associated to each product j that is really used in the marketing campaign.

Since this problem belongs to the class of NP-hard problems, the development of an efficient technique for solving the TODMC is of great importance. Thus, in this present work a new hybrid algorithm, named GGVNS, is proposed for solving TODMC. It is inspired by the metaheuristics Greedy Randomized Adaptive Search Procedure – GRASP [4] and General Variable Neighborhood Search – GVNS [7], which uses the Variable Neighborhood Descent –VND [8] as local search strategy. The ability of the GRASP to produce a set of feasible and good quality solutions was used to generate a first solution for the GVNS procedure. GVNS was chosen due to its simplicity, efficiency and its natural search ability to handle with different neighborhoods [12].

2 Methodology

2.1 Solution representation and evaluation

A solution is represented by a binary array $R_{|C| \times |O|}$, where C indicates the set of available costumers and O represents the possible products to be used in the campaign. If a given cell $s_{i,j} | i \in C, j \in O$ is equal to "1" (true), the product j will be offered to the client i; otherwise, the value would be "0" (false). The goal is to maximize the evaluation function f, given by Eq. (1), which is composed of two parts: f^{obj} and f^{inv}.

(1)
$$f(s) = f^{obj}(s) + f^{inv}(s)$$

The part f^{obj} refers to the problem objective function itself and it is calculated by Eq. (2):

(2)
$$f^{obj}(s) = \sum_{i \in C} \sum_{j \in P} (r_{i,j} - d_{i,j}) s_{i,j} - \sum_{j \in P} f_j y_j$$

The second part, f^{inv}, is given by Eq. (3), and it is associated with the attending of the operational constraints of the problem.

$$(3) \qquad f^{inv}(s) = \sum_{i \in C} \left(f_i^M(s) \right) + \sum_{j \in P} \left(f_j^O(s) + f_j^B(s) \right) + f^R(s)$$

In Eq. (3), $f_i^M(s)$ evaluates s regarding to the attending of the client maximum offers (M_i), since a given client i should not receive more offers than he expects. On the other hand, $f_j^O(s)$, $f_j^B(s)$ and $f_j^{HR}(s)$ are related to the active products quoted to be used in the campaign, which measures the following requirements: $f_j^O(s)$ checks a solution s regarding to the fact that the product j had been offered to smaller amount of clients than O_j^{min}. $f_j^B(s)$ checks s if the available money wallet (budget B_j) for each product was respected. Finally, $f^R(s)$ checks if minimum expected campaign profit (campaign hurdle rate (R)) was achieved.

2.2 Building an initial solution

This paper presents a novel greedy randomized solution generator, adapted from the greedy procedure proposed by van Praag [11] and, also, used by Nobibon et al. [10].

The proposed procedure is mainly based in the weighting of the "best" set of costumers able to receive an offer. Greedy parameters γ were inserted in way to select randomized clients sorted by the $NPP_{ij} = \frac{r_{ij} - d_{ij}}{d_{ij}}$. If γ is equal to "1" the selection becomes total greedy (the offers s_{ij} with the higher NPP_{ij} are selected first). On the hand, if this value is equal to "0" it becomes totally random.

2.3 Neighborhood structures

To explore the solution space of the problem, three Neighborhood Structures – NS, similar to those proposed by Nobibon et al. [10], were developed.

It is necessary to emphasize that all movements described and used in this work keep the feasibility of the solutions. A brief description of the NS is presented below.

Swap Clients Intra - $N^{SC-Intra}(s)$: This movement consists in swap two positions, $l, m \in C$ of a given product $j \in P$, such that $s_{l,j} = s_{m,j}$ and $s_{m,j} = s_{l,j}$, in a way that the operational constraints of the problem remain respected.

Swap Clients Inter - $N^{SC-Inter}(s)$: Similar to the movement $N^{TC-Intra}(s)$,

but this one can be applied in the swap of different products $i, j \in P$, e.g., $s_{l,i} = s_{m,j}$ and $s_{m,j} = s_{l,i}$.

Swap Products - $N^{SP}(s)$**:** exchanges two different columns $i, j \in P$ such that $y_i = 1$ e $y_j = 0$ or unlike. In these movements, a product that was not being used in the campaign ($y_j = 0$) can be now part of the active set of products ($y_j = 1$) used in the campaign.

2.4 Proposed Algorithm

The proposed algorithm, so-called GGVNS, combines the procedures GRASP and GVNS. The pseudo-code of the GGVNS algorithm is outlined in the Algorithm 1. In this algorithm, *GRASPMaxIter* represents the maximum number of solutions generated by the GRASP procedure and *IterMax* indicates the maximum number of iterations performed at a given level of perturbation.

Algorithm 1 `GGVNS`

Input: *GRASPMaxIter* and *IterMax*, Function $f^{obj}(.)$
Output: Final solution s^*

1:   $s_0 \leftarrow$ Best solution in *GRASPMaxIter* iterations of the procedure greedy procedure (Section 2.2), $s^* \leftarrow$ VND(s_0, f) and $p \leftarrow 0$
2: **while** stopping criterion not satisfied **do**
3:    $iter \leftarrow 0$
4:    **while** $iter < IterMax$ **and** stopping criterion is not satisfied **do**
5:      $s' \leftarrow$ Refine(s^*, p, f)
6:      **if** s' it is better than s^* according to the function f **then**
7:        $s^* \leftarrow s'$; $p \leftarrow 0$ and $iter \leftarrow 0$
8:      **else**
9:        $iter \leftarrow iter + 1$
10:      **end if**
11:    **end while**
12:    $p \leftarrow p + 1$
13: **end while**
14: return s^*

An initial solution s_0 (line 1 of the Algorithm 1) is created by applying the GRASP construction phase, described in Section 2.2, during *GRASPMaxIter* iterations.

When the current solution s' is better than the s^*, the algorithm resets the level counter and the iteration counter. The local search is made by the

classical VND procedure [8], using the NS described in the section 2.3 in random order (line 2 of the Algorithm 2). The "Shaking" procedure, line 4 of the Algorithm 1, performs the application of a random move of the NS list.

Algorithm 2 `Refine`

Input: r neighborhoods in random order, level p, solution s and function f^{obj}
Output: Final solution s

  1: **for** $i \leftarrow 1$ to $p + 2$ **do**
  2:     $r_{neigh} \leftarrow$ neighborhood in r chosen at random
  3:     $s' \leftarrow$ Shaking(s, r_{neigh})
  4:     $s \leftarrow$ VND(s', f)
  5: **end for**
  6: return s

---

3   Computational experiments and discussions

The tests were carried out on a OPTIPLEX 9010 Intel Core i7-3770, 3.40 $\times$ 8 GHZ with 32 GB of RAM, with operating system Ubuntu 12.04.3 precise, and compiled by g++ 4.6.3 using the Eclipse Kepler Release.

To validated the proposal, 4 subgroups of instances were considered: small (S3), medium (M1 and M2) and large (L), each one with different number of clients. Each subgroup contains 3 others subgroups (composed with different number of product offers: 5, 10 and 15) with 18 instances each, which differ with respect to the profits, costs and boundaries of the operational constraints.

Three set of tests were realized. The first one was realized in order to verify the power of the proposed GRASP to build good quality initial solutions. The second set was made with the same parameters as the first one; however, the local search procedure VND was called after each solution created by the GRASP. Finally, in the last set of tests the complete proposed algorithm, GGVNS, was applied. It performs its search over the GRASP best solution (as described in Algorithm 1).

In each set 216 executions were made. After some preliminary empirical analysis, $IterMax$ was set in 130 and GRASPMaxIter in 100000 solutions.

The best known results, so-called f_i^*, were provided by Nobibon et al. [10]. To evaluate the performance of the proposed algorithm, the GAP metric was used, with $gap_i^n = \frac{f_i^* - f_i^n}{f_i^*}$. In this metric, f_i^* the best known result for each benchmark problem i and f_i^n the value obtained by each algorithm.

Table 1 presents the computational results of the three set of tests. Column

"n" indicates the method. Columns "Init." and "H8" are related to the values presented by the greedy construction algorithm and Tabu Search of Nobibon et al. [10].

Table 1
Computational results – GAP (%)

n	S3			M1			M2			L		
	5	10	15	5	10	15	5	10	15	5	10	15
Nobibon et al. results [10]												
Init.	19.20	17.85	18.34	17.90	16.20	18.36	18.92	19.23	17.84	26.22	24.86	25.65
H8	6.86	6.52	7.76	**7.22**	8.54	**7.60**	9.75	**9.58**	**9.11**	**10.86**	**11.04**	10.23
γ	Set of tests I – Pure GRASP											
0	19.20	17.85	18.34	17.90	16.20	18.36	18.92	19.23	17.84	26.22	24.86	25.65
0.2	16.12	14.55	17.14	17.44	14.52	16.31	17.12	17.32	16.32	24.98	23.32	23.98
0.4	15.25	13.89	18.98	15.22	13.32	17.56	17.54	18.76	15.56	24.33	22.56	23.67
0.6	12.55	16.25	17.76	15.96	13.76	16.87	17.95	17.85	15.74	23.91	22.16	23.18
	Set of tests II – GRASP + VND											
0	19.11	15.76	17.10	15.90	14.20	15.96	16.31	16.30	13.00	22.86	20.32	22.31
0.2	15.54	12.34	16.54	15.44	14.43	15.25	15.98	16.72	13.10	19.54	19.54	19.43
0.4	14.22	12.54	16.54	15.00	11.13	14.13	15.31	15.13	12.95	20.56	20.98	20.98
0.6	11.32	11.87	16.76	12.14	10.15	13.33	16.45	15.52	13.03	20.98	20.10	18.76
	Set of tests III – GGVNS											
0	10.68	6.98	7.77	9.72	9.12	12.15	11.29	9.90	11.15	12.99	12.29	12.15
0.2	6.90	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	9.10	7.89	**9.68**	9.76	9.91	**10.86**	11.22	10.11
0.6	**6.77**	6.50	7.49	7.24	**8.42**	7.63	9.85	9.78	9.44	11.51	11.93	**10.22**

By analyzing Table 1, it is noteworthy that the proposed GGVNS algorithm was able to obtain lower or equal GAPs compared to those presented by Nobibon et al. [10], arching lowest values in six groups of instances. The randomized greedy construction method was validated, since it presents better values in all $\gamma \neq 0$. This result is entirely consistent, in view of the potential of GRASP method in reaching different attraction basins.

As extensions of this work, the development of new constraints is proposed, like non-linearity involving the costs when big amount of products are being offered. Thus the cost of offering the product j could vary as the number of clients increases, for example, it can becomes each time more cheaper.

Similarly, customers who receive more than one offers could have a some operational costs reduced. Such restrictions are entirely consistent and represent real cases that happen in day-to-day operations of product offering campaigns. In the computational aspect, the development of new neighborhoods structures and shaking strategies are proposed.

References

[1] Bhaskar, T., R. Sundararajan and Krishnan, P. G., *A fuzzy mathematical programming approach for cross-sell optimization in retail banking*, Journal of the Operational Research Society **60** (2009), pp. 717–727.

[2] Chun, Y. H., *Monte Carlo analysis of estimation methods for the prediction of customer response patterns in direct marketing*, European Journal of Operational Research **217** (2012), pp. 673–678.

[3] Cohen, M. D., *Exploiting response models: optimizing cross-sell and up-sell opportunities in banking*, Informations Systems **29** (2004), pp. 327–341.

[4] Feo, T. A. and M. G. C. Resende, *Greedy randomized adaptive search procedures*, Journal of Global Optimization 6 (1995), pp. 109–133.

[5] Glover, F. and Laguna, M., *Tabu Search*, Kluwer Academic Publishers (1997).

[6] Guadagni, P. M. and J. D. C. Little, *A logit model of brand choice calibrated on scanner data*, Marketing Science **3** (1983), pp. 203–238.

[7] Hansen, P., N. Mladenovic and J. A. M. Pérez, *Variable neighborhood search: methods and applications*, 4OR **6** (2008), pp. 319–360.

[8] Mladenovic, N. and P. Hansen, *A variable neighborhood search*, Computers & Operations Research **24** (1997), pp. 1097–1100.

[9] Moro, S., M. S. R. Laureano and P. Cortez, *Enhancing bank direct marketing through data mining*, CAlg European Marketing Academy (2012) pp. 1–9.

[10] 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.

[11] van Praag, N., *Optimization of promotion campaigns using tabu search*, Master's thesis, Faculty of Business and Economics, KULeuven (2010).

[12] Souza, M. J. F., I. M. Coelho, S. Ribas, H. G. Santos and L. H. C. Merschmann, *A hybrid heuristic algorithm for the open-pit-mining operational planning problem*, European Journal of Operational Research **207** (2010), pp. 1041–1051.