IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 December 2023, accepted 4 January 2024, date of publication 11 January 2024, date of current version 13 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3353052

RESEARCH ARTICLE

GATeS: A Hybrid Algorithm Based on Genetic Algorithm and Tabu Search for the Direct Marketing Problem

FELIPE MARTINS MÜLLER[1,2,3], VANESSA A. SCHNEIDER[4], IAÊ SANTOS BONILHA[4], VERIDIANE BARBARA DE SOUZA[4], GREICI DA ROSA DA CRUZ[4], AND MICHALIS MAVROVOUNIOTIS[5]

[1]Department of Applied Computing, Federal University of Santa Maria, Santa Maria, Rio Grande do Sul 97105-900, Brazil
[2]PPGEP, Federal University of Santa Maria, Santa Maria, Rio Grande do Sul 97105-900, Brazil
[3]PPCC, Federal University of Santa Maria, Santa Maria, Rio Grande do Sul 97105-900, Brazil
[4]Department of Postgraduate Production Engineering, University of Santa Maria, Santa Maria, Rio Grande do Sul 97105-900, Brazil
[5]ERATOSTHENES Centre of Excellence, 3012 Limassol, Cyprus

Corresponding author: Felipe Martins Müller (felipe@inf.ufsm.br)

The work of Felipe Martins Müller was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES) under Finance Code 2380-14-5. The work of Michalis Mavrovouniotis was supported in part by the EXCELSIOR: ERATOSTHENES: Excellence Research Centre for Earth Surveillance and Space-Based Monitoring of the Environment H2020 Widespread Teaming Project (www.excelsior2020.eu) through the European Union's Horizon 2020 Research and Innovation Program from the Government of the Republic of Cyprus through the Directorate General for the European Programs, Coordination and Development and the Cyprus University of Technology under Grant 857510.

ABSTRACT This paper deals with the problem of selecting a set of clients that will receive an offer of one or more products during a promotional campaign. Such campaigns are essential marketing tools to improve the economic profit of an enterprise, either by acquiring new customers or generating additional revenue from existing customers. In this research, a well-known mathematical model for the problem is used and extended with the cannibalism constraint which avoids some products being offered simultaneously to simulate competing products cannibalizing each other's market. To solve this problem, a hybrid heuristic is proposed, which uses a genetic algorithm (GA) as long-term memory for a tabu search (TS). The main idea is not to use GA exclusively as an optimization procedure but also as a diversification strategy. In particular, GA elite solutions replace the TS's current solutions exploring in this way new areas in the search space. GA also receives the best TS solutions to maintain its population with high-quality solutions. Extensive computational experiments are performed on a set of existing benchmark test problems integrated with the restriction of cannibalism. A new set of instances with a high degree of difficulty is generated and are available to the research community through GitHub. The proposed method is compared with state-of-the-art methods demonstrating better overall performance (sometimes more than 10 percentage points) and statistical significance.

INDEX TERMS Cannibalism, direct marketing problem, genetic algorithm, hybrid algorithm, metaheuristics, tabu search.

I. INTRODUCTION

A huge challenge of direct marketing is how to 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 consumers, the market, the economy, demography, technology, politics, and sociology that help data analysts make their decisions.

Promotional campaigns are one of the most direct marketing fundamental tools for client acquisition and overall profit generation [1], [2], [3], [4], [5], [6]. These campaigns

The associate editor coordinating the review of this manuscript and approving it for publication was Huaqing Li.

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 complex combinatorial problems that are difficult to solve and deserve more attention from the research community [7], [8]. For more details on relevant problems, please refer to the following paper [9].

There are historical concepts related to the evolution of customer management over the last two decades that underpin promising directions for future customer management research [10]. Key trends and perspectives confirm the dynamic evolution of the market and the notable decline in the effectiveness of traditional mass marketing approaches [11]. The investigation of more assertive optimization strategies in the context of marketing was explored using different problem-solving methods, such as matheuristics, statistics, and evolutionary computation. Recent marketing studies have considered issues related to the processing of customer data, grouping, or segmenting information to increase the positive return on direct marketing campaigns [5], [12], [13], [14]. However, it is important to call attention that in the field of optimization related to direct marketing, to the best of the author's knowledge, the cannibalism strategy was never tackled in existing works.

Heuristic approaches tend to be promising solutions to address direct marketing problems (DMP) as a substitute for statistical methods [3], [5], [15]. Some examples can be found in [16] who proposes the use of fuzzy logic for client selection in a cross-sale marketing campaign from a bank. Cohen [15] aggregates customers into groups; then optimizes the number of customers who receive certain products within each group; and ultimately assigns products to individual customers using an assignment model for each group.

Nobibon et al. [3] proposed a mathematical formulation for the DMP as well as two main approaches to solve it: an exact approach and a Tabu Search (TS) heuristic, which performed well in small and large-scale instances, respectively. They also made available a set of test instances and bounds for the problem that will be used later in the experiments. Oliveira et al. [17] have presented a hybrid scheme with GRASP (Greedy Randomized Adaptive Search Procedure) and VNS (Variable Neighborhood Search).

Cetin and Alabas-Uslu [9] proposed a different approach that divides DMP into two decision problems: 1) assigning products to the market campaign and 2) assigning offers to the client base. The decision problems were solved with linear programming and a heuristic connection between them, obtaining good results for all types of instances. However, since an exact procedure is used, there are limitations as the size of the problem grows. In the experiments later on we demonstrate that there are some instances that those approaches were unable to solve.

Recently [5] presented a matheuristic to solve the DMP in a telecommunications environment. They included some business and customer-specific constraints, including conflict management, and presented a good literature review about the latest improvements in the field.

Our research started with [18] and [19], where we proposed two hybrid schemes combining GRASP and Genetic Algorithm (GA) with Tabu Search, respectively. Then, according to the best of our knowledge, for the first time, the cannibalism constraint was considered in the direct marketing problem [18]. Cannibalism between products can be defined as the possibility of the sale of a destructive product interfering with the sale of a similar product [20], imposing more challenges to the problem. Since DMP with cannibalism is a special case of DMP which is strongly NP-hard, it can be reduced to a generalized assignment problem that is known to be NP-hard [21]. Later [22] has adapted a mathematical model to solve almost all instances generated so far, including cannibalism. Coelho et al. [23] extended the VNS proposed by [17], to a bi-objective version of the problem based on the concepts of maximizing profits and, at the same time, searching for a set of customers with less variability over their expected return.

In this context, the present study seeks to answer the research question: "How to optimize the selection of customers and products in direct marketing campaigns, considering the cannibalism between products, using heuristic approaches, to maximize the net profit generated by the campaign offers?". Thus, the innovative hybridization of GA and TS (GATeS) proposed in this work emerges as a promising approach to solve the challenge of cannibalism between products in direct marketing campaigns. The main motivation of this research is therefore to exploit this gap in knowledge and offer an innovative approach to deal with inter-product cannibalism in direct marketing campaigns.

In this paper, GATeS, a TS with an embedded GA, to address the DMP with the cannibalism constraint is proposed. A modified mathematical model that includes the cannibalism constraint is also proposed. Experiments between GATeS and state-of-the-art approaches [3], [9] are performed in the existing benchmark instances as well as in newly generated large benchmark instances. All source codes to run simulations, datasets, and results presented in this work are publicly available at a GitHub repository [24] under the MIT license. In summary, the contributions of this work are as follows:

- Innovative strategy to combat product cannibalism in direct marketing campaigns.
- A novel and robust hybrid heuristic based on GA and TS (GATeS), in which the GA works as a long-term memory for the TS procedure.
- New sets of benchmark instances with high degree of difficulty have been made available (i.e., Group 3 in the experiments).
- Datasets and codes made available to the research community, driving the continuous advancement of knowledge in DMP.

The remainder of this paper is organized as follows. Section II describes our research steps dealing with DMP. Section III presents the DMP, its mathematical model, and a brief description of the other heuristic procedures compared to GATeS. Section IV describes the hybrid algorithm GATeS and all procedures related to it. Section V presents the experimental setup and the way it generates a new set of instances and VI presents and discusses the results. Section VII provides the data repository and the last section (VIII) concludes the paper showing potential future works.

II. RESEARCH STEPS

This research started by dealing with the DMP in 2016, after an extensive bibliography review identified that many practical situations were not taken into account. Then was decided to investigate the introduction of cannibalism in the DMP. Since [3] provides a set of instances with the best solution so far for each of them, their heuristic approaches were adapted to deal with cannibalism. This investigation works in the set of instances generating the cannibal pairs using Euclidean distance analysis. The cannibal pair generation used the ratio of one pair for every 5 products offered, which allowed us to continue finding feasible solutions in the same set of instances. Then a Tabu Search was implemented using GRASP as an initial solution including a neighborhood that allows changes in the number of offers per product. Comparisons were made against the original TS and also against the VNS proposed by [17] in selected instances published by them without cannibalism. These results were published in 2018 by [18] and showed that DMP considering cannibalism is more difficult to solve and heuristics could be a promising approach to solve it.

Continuing the studies, two other ways were proposed to generate cannibal pairs called similarity and dissimilarity, based on the frequency of the clients assigned to receive offers. Then the lower bounds were updated for all instances in the data set. Proving once more that cannibalism makes the DMP more difficult to solve. These results were published in 2019 by [22]. In the same period was possible to see how GA could improve the performance of TS, the preliminary results can be found in [19].

Meanwhile, it was decided to implement and adapt to deal with cannibalism the matheuristic proposed by [9], this procedure was the one with the best performance so far but did not achieve the solution for some instances. That weakness was never commented on in the published papers nor in the original thesis. Also, it was proposed a new set of instances from 40 up to 100 thousand clients, and was proved that matheuristic fails in more cases and that applying heuristics is a suitable approach. These results were published in 2020 by [25].

Finishing this cycle been presented GATeS, which uses all the knowledge developed by our research group in order to solve the DMP with the cannibalism constraint. Gates uses TS as an optimizer and GA as an evolutionary repository of elite solutions, occasionally providing new starting solutions to TS. That hybridization creates a very good way to escape from the local optimal traps in a solution space flat containing lots of similar good solutions. Therefore, there was no reason to be worried about the performance of GA since it includes diversity in TS working as a long-term memory.

III. DIRECT MARKETING PROBLEM

A. PROBLEM DESCRIPTION

The DMP is divided into two decision problems. The first problem is to select which products will be included in the promotional campaign. Recall, that promotional campaigns usually focus on a group of clients and have to be tailored to avoid saturating clients with offers. Therefore, the selected products must have a strong impact on the campaign outcomes. The second decision problem is when clients receive offers for the products included in the promotional campaign. These two decision problems can be represented by two binary variables: $y_j = \{0, 1\}$, which indicates whether the product j is participating in the market campaign or not; and, $x_{ij} = \{0, 1\}$, 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.

The model proposed by [3] is adapted to cope with cannibalism as shown in [22]. The DMP is comprised of two main elements: the client set C and the product set P. Each product $j \in P$ has a budget B_j; a fixed cost f_j which is the one of the product j participating in the campaign; and an offer quota O_j, which is the minimum number of clients that must receive the offer to make its participation justifiable in the marketing campaign. Each client $i \in C$ has a projected profit p_{ij} for each product offer j; a cost c_{ij} associated with each product offer j to a client i; a net potential profit $NPP_{ij} = (p_{ij} - c_{ij})/c_{ij}$, which represents the return per monetary unit invested in an offer of the product j to a client i; and a limit M_i which is the limit that is placed to simulate offer saturation that could result in a clients' negative response towards the campaign.

The mathematical model can be defined as follows:

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

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

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

$$\sum_{j=1}^{n} x_{ij} \leq M_i, \quad i = 1, \ldots, m \tag{4}$$

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

$$\sum_{i=1}^{m} x_{ij} \leq my_j, \quad j = 1, \ldots, n \tag{6}$$

$$y_i + y_j \leq 1, \forall (i,j) \in CAN \tag{7}$$

$$y_j, \; x_{ij} \in \{0,1\} \quad i = 1, \ldots, m, \; j = 1, \ldots, n \tag{8}$$

The main goal of DMP is to maximize the net profit generated by all the offers made in the campaign; see equation (1). To be considered viable, the market campaign must achieve the hurdle rate H, the return per monetary point invested in the campaign; see equation (2). Equation (3) ensures that the maximum budget per product is respected. Note that the inclusion of 0-1 variable y_j in the right-hand side of the equation tightens the relaxed model and consequently speeds up any commercial solver that makes use of the information from the relaxed model. 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. 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 indexes belong to the set CAN, then they cannot be offered to the clients in the same campaign. Ultimately, the last constraint in equation (8) is the integrality constraint. Observe that if the equation (7) is withdrawn, then the model is the same as in [3].

In short, the DMP tries to assign m clients to m products, satisfying the budget constraints (B_j) per product, the maximum offers per client (M_i), the minimum clients per product (O_j), in order to maximize the net profit. An example of DMP is presented in the following section.

B. ILLUSTRATIVE EXAMPLE

To better understand the problem and the algorithms, an illustrative example comprising 10 clients, 5 products, a hurdle rate (H) of 5%, and the cannibal pair (1,4) is presented. As a result, products 1 and 4 can not be offered simultaneously in the same campaign. Table 1 presents the cost of offering each product to each client. Table 2 presents the profit that each client generates if the offer is accepted. Table 3 presents the complementary input data. Note that, these data will be used in further examples.

Figure 1 represents a feasible solution that is been obtained by the constructive algorithm (described in Section IV-A later on). The products are represented by $j_i, i = 1, \ldots, 5$, followed by ten circles. The number inside the circles corresponds to the potential clients to be assigned in a non-increasing order by NPP_{ij}. Filled circles indicate that the client is assigned to the product; otherwise empty circles. It is possible to observe that products 1, 3, and 5 are offered and the cannibalism constraint is satisfied (recall, products 1 and 4 are not offered simultaneously). Products 2 and 4 are not offered. The constraints in equations (2) to 7 are satisfied and the objective function value is 106.

TABLE 1. Cost of offer products to clients.

	Clients									
Products	1	2	3	4	5	6	7	8	9	10
1	**1**	**2**	**3**	**2**	**3**	**3**	**2**	**1**	**2**	**3**
2	**3**	**2**	**1**	**1**	**3**	**1**	**3**	**3**	**1**	**1**
3	**1**	**1**	**1**	**1**	**3**	**2**	**2**	**3**	**2**	**1**
4	**3**	**3**	**2**	**1**	**1**	**1**	**2**	**2**	**2**	**1**
5	**1**	**1**	**2**	**1**	**2**	**3**	**3**	**1**	**1**	**1**

TABLE 2. Profit of clients per product.

	Clients									
Products	1	2	3	4	5	6	7	8	9	10
1	**15**	**12**	**10**	**7**	**11**	**1**	**14**	**11**	**4**	**5**
2	**4**	**5**	**12**	**0**	**5**	**1**	**16**	**16**	**9**	**7**
3	**13**	**11**	**15**	**11**	**10**	**10**	**11**	**16**	**11**	**5**
4	**12**	**10**	**16**	**3**	**13**	**14**	**16**	**12**	**10**	**1**
5	**7**	**6**	**14**	**5**	**13**	**13**	**2**	**6**	**16**	**16**

TABLE 3. Complementary data.

	Index									
Data	1	2	3	4	5	6	7	8	9	10
M_i	**3**	**3**	**2**	**2**	**2**	**2**	**3**	**2**	**2**	**3**
O_j	**5**	**8**	**8**	**6**	**7**					
B_j	**12**	**20**	**15**	**21**	**26**					
f_j	**23**	**47**	**35**	**42**	**59**					

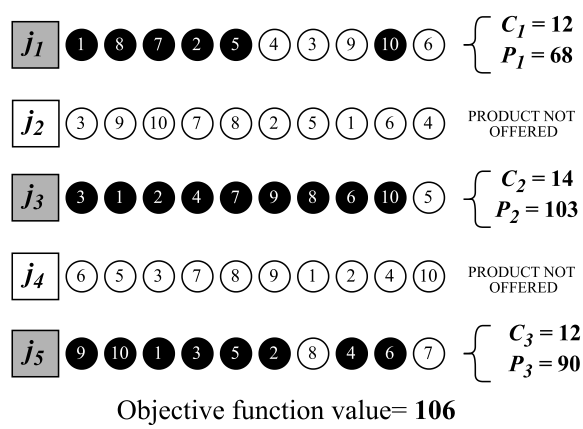


FIGURE 1. Feasible solution obtained by the constructive algorithm 1.

C. ADDRESSING THE DIRECT MARKETING PROBLEM

Nobibon et al. [3] presented seven heuristics and a TS algorithm, that outperformed existing methods used at that time. They also provided a set of instances with the best solution value found so far by a branch-and-price algorithm.

Two heuristics are presented in [9] to solve DMP, 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 the first phase, the heuristic rules were used to predict which products would be selected for or removed from the product campaign and then distribute these products to the customers optimally. The two aforementioned phases are connected via a heuristic rule. Two alternative heuristic rules, derived from the proposed LP model in the first phase, called HR-1 and HR-2, are suggested to predict the products eliminated from the campaign (or equivalently the products involved in the campaign).

Specifically, HR-1 rule takes the optimal solution of the modified model where each variable y_j is replaced by a dummy variable $x_{m+1,j}$ and then $x^*_{m+1,j}O_j$ is treated as excluding the possibility of product j. Then HR-1 rule singled out the products that have the highest exclusion possibility until the total amount O_j of eliminated products meet the following inequality:

$$\sum_{j \in N_c} O_j \geq C, \tag{9}$$

where N_c is the set of excluded products and $C = \sum_{j=1}^{n} O_j - \sum_{i=1}^{m} M_i$, is a constant.

In contrast to HR-1 which runs a new LP for each product elimination, HR-2 uses the duality properties to devise which products will be discharged without additional computational efforts. Hence, the second rule suggests that a higher ratio of (f_j/B_j) for product j is closer to being eliminated from the promotion campaign. The termination condition of HR-2 is the same as with HR-1 which is based on equation (9). After that, an assignment problem has been executed to place the clients into the remaining products. These heuristic rules generate two different procedures that outperform the results presented by [3]. However since commercial solvers are used to run the associated LP for the heuristic rules, they have a limited capacity in the problem dimension to be solved. This last issue is not demonstrated in the original paper, but some unsolvable instances by HR-1 or HR-2 are presented in the experiments later on.

Oliveira et al. [17] presented a hybrid heuristic algorithm, based on the Greedy Randomized Adaptive Search Procedures (GRASP) [26] and General Variable Neighborhood Search (VNS) [27] to address DMP. These authors' approach outperformed the results of [3], but it was outperformed by the results of [9]. This is probably because their research was published in the same year. Once the VNS was surpassed by HR access to the VNS codes was not possible. This paper only shows the results involving the instances that were published in the original articles compared against GATeS.

A TS based on [3] is also implemented for the experiments, including a neighborhood that moves a client to a different product and uses a GRASP to generate a different initial solution at each restart [18]. Comparisons are made between these four heuristics, i.e., GATeS, TS, HR-1, and HR-2.

D. CANNIBALISM BETWEEN PRODUCTS

From the insertion of the cannibalism constraint in the model of [3], the offer of similar products is prevented making them mutually exclusive. Thus, when one product of the cannibal pair (j_1, j_2) receives a positive response (i.e., $y_{j_1} = 1$), then the other product can not be offered (i.e., $y_{j_2} = 0$). According to [28], cluster analysis is popular in many fields, including marketing segmentation, identifying subgroups of customers with a similar profile, and who may be receptive to a particular form of advertising. The present study explores three different strategies to deal with these cannibalistic product pairs.

The first strategy was proposed by [18], uses statistical methods, such as Euclidean distance [29], in conjunction with the single linkage clustering method [30] to define cannibalistic pairs. This approach analyzes the simulated profit per customer selecting the closest variables to determine the cannibal pairs by taking into account the number of products in each instance.

The second and third strategies introduce new criteria to identify cannibal pairs, named "Dissimilarity" and "Similarity". Initially, the exact solutions to the original problem without cannibalism were analyzed to identify which products were actually offered and to which customers the offers were been made.

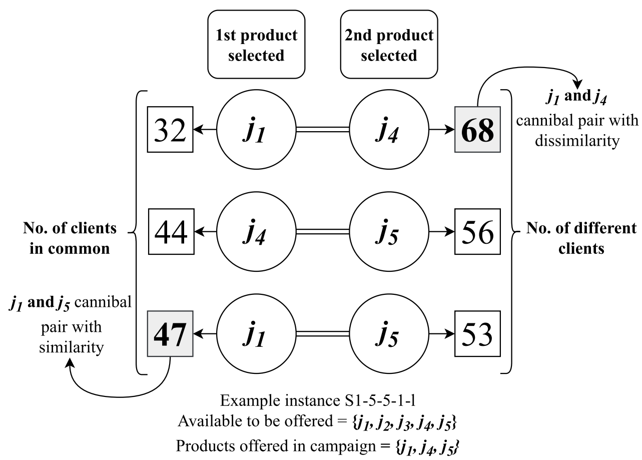


FIGURE 2. Representation of cannibal pair formation for similarity and dissimilarity.

Dissimilar product pairs could be identified considering the products offered that had the highest number of offers to different customers. Similar product pairs also could be identified based on the products offered to the largest number of equal customers. Figure 2 presents an illustrative scheme on how the combination of pairs occurs for instance S1-5-5-1-l and extends the same logic to all instances updated with the new criteria.

Instances will have one cannibal pair out of five candidate products, i.e., instances with 5 products will have 1 cannibal pair, with 10 products 2 pairs, and so on. The same instance will not have identical cannibal pairs. The instances that [3] proposes were updated and a new set of large and huge instances were generated.

The introduction of cannibalism between products is an important factor in decision-making, representing a breakthrough in the direct marketing literature, and also offers promising potential to drive successful outcomes in terms of return on investment and profitability.

IV. HYBRID ALGORITHM: GATES

GATeS is a hybrid algorithm of TS with a GA and, to the best of our knowledge, a GA was never used as a long-term memory in a TS search procedure. The proposed hybrid algorithm aims to balance the exploration and exploitation tasks in order to address the DMP. All procedures that have been developed are presented in detail in the following subsections as well as the way they interact.

A. CONSTRUCTIVE ALGORITHMS

To build the initial population of GA three different ideas for constructive algorithms are used: two deterministic and one stochastic. The first constructive algorithm is based on the initial solution procedure of the TS algorithm proposed in [3] with modifications to the product selection procedure. The second constructive algorithm is a greedy randomized version of the first one. Lastly, the third constructive algorithm starts with an infeasible solution, and a repairing procedure is applied to ensure the feasibility.

1) CONSTRUCTIVE ALGORITHM BASED ON PROFIT AND COST

This constructive algorithm follows the same offer set construction logic proposed by [3], based on NPP_{ij}, with an additional procedure for product inclusion, and the ability to handle cannibalism (See Algorithm 1). Through the rule that prioritizes higher profit offers, there is a possibility of feasible offer sets to be disregarded, especially when the budget is tight. For example, if a product has a low budget and the majority of its most profitable offers have a high cost, then the resulting 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}. Afterward, the algorithm tries to improve the offer by replacing the client with minimum NPP_{ij} in the candidate offer with a client with maximum NPP_{ij}, among those available, while there is a sufficient budget. If a feasible offer set is found with a higher profit than the original one, it is then replaced and that offer is included in the campaign. From Figure 1 is possible to see that this last step cannot be applied because there are not enough customers available to complete a feasible offer. After this step, the algorithm tries to allocate profitable remaining customers while the solution remains feasible. As it can be attended, two constructive algorithms are generated.

Algorithm 1 Constructive Algorithm Adapted From TS by [3]

Initial assignment based on profit.

1: Sort by non-increasing order of NPP the available clients for each product.
2: Allocate O_j best-ranked available clients to the product that produces the highest return respecting the problem constraints.
3: Include the product in the campaign and update the set of available clients.
4: Repeat the previous three steps until there are no more products possible to offer.

Assignment of remaining products based on cost.

1: Sort by non-decreasing order of c_{ij} the available clients for each product not yet in the campaign.
2: Allocate O_j best-ranked available clients to the product that produces the highest return respecting the budget constrain.
3: Swap the allocated client with minimum NPP with the available client with maximum NPP while there is enough budget.
4: If the product produces a feasible offer include it in the campaign and update the set of available clients.
5: Repeat the previous four steps until there are no more products possible to offer.

Assignment of remaining profitable clients.

1: Include into the products belonging to the campaign all available clients that can improve the profit respecting the problem constraints.

2) GREEDY RANDOMIZED CONSTRUCTIVE ALGORITHM

The second idea was a greedy randomized version of the first algorithm, that is, instead of selecting offers in decreasing order of profitability, a random offer among the 10% best offers available is selected. In the preliminary experiments, it could be observed that some profitable customers were not appearing in the product offerings. Therefore, a variant of the greedy random algorithm is designed that instead of limiting the number of customers allocated in the product j by the minimum number of offers O_j, allows the inclusion of customers as long as there is a budget B_j. These two variants are the basis for generating the GA's initial population. At every greedy randomized algorithm run products may be blocked randomly, improving population diversity. Solutions obtained by deterministic constructive algorithms are also included in the GA initial population.

3) CONSTRUCTIVE ALGORITHM WITH SOLUTION REPAIR PROCEDURE

The third idea is also deterministic but the difference with the previous approaches that start with an infeasible solution and applies a repair procedure until the solution becomes feasible. This algorithm starts with an offer set for each product that can extrapolate at most five times each product budget or contain all available clients, ignoring other constraints. After each offer set is produced, the algorithm tries to repair them by analyzing the infeasible offers in decreasing order of profitability. The profitability with different degrees of importance is considered, i.e., the viable (most profitable clients under budget) part is counted in full while only 20% of the profitability of unfeasible offers (most profitable clients over budget) counts. This prevents offers with many clients above the budget from being evaluated first. The repair process tries to eliminate conflicts among clients that violate the maximum number of offers allowed as shown in 2.

After each offer set is produced, the algorithm tries to repair each product offer set in decreasing order of profitability of the infeasible offer sets. Higher importance to the feasible part of the offer set is given, 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 the offer limit. This repair will be done by replacing the conflicting offer of one of the products in a way that no other constraints of the problem will be violated. (see Algorithm 3).

Algorithm 2 Repair Based Constructive Algorithm

1: Sort by non-increasing order of *NPP* the available clients for each product.
2: For each product allocate the best-ranked available clients until there are no more clients to allocate or the offer cost reaches five times the budget. Split the allocated clients into two sets: S_1 contains the first allocated clients respecting the budget and S_1 contains allocated clients over budget.
3: Sort the products in a non-increasing order by a weighted net profit composed of the return from the clients belonging to S_1 and 20% of the return from the clients belonging to S_2.
4: Apply the **Repair Procedure (Algorithm 3)** for each product considering the previous rank.
5: Include into the products belonging to the campaign all available clients that can improve the profit respecting the problem constraints.

4) INITIALIZING GENETIC ALGORITHM AND TABU SEARCH

The four aforementioned constructive algorithms are used as follows. TS runs 8 times for each instance, the first time TS takes as the initial solution, the one obtained by Algorithm 1 described in section IV-A1, in the second time, the one obtained by the repair-based algorithm is used as the initial solution (Algorithm 2) this procedure is been described in section IV-A3, in the third time it is used as the initial solution the one obtained by the variant of the greedy randomized algorithm, the rest of the time it uses as initial solution those obtained by the greedy randomized algorithm, both are been described in section IV-A2.

GA initializes its first population (containing 100 individuals) using the greedy randomized algorithms described in section IV-A2. Some changes are allowed in the order of product selection as well as some products can be blocked at random increasing in this way the population diversity. The new generation of individuals includes the solutions generated by the constructive algorithms and the greedy randomized algorithms as well as the best solution so far (i.e., elitism is enabled).

Algorithm 3 Repair Procedure - Function for Managing Offer Conflicts

1: Disregard the product that has S_1 with fewer clients than O_j or its cannibal pair is already in the campaign otherwise try to repair the offer.
2: Analyse each client in S_1 in the candidate product. If the client does not reach M_i offers keep it otherwise try to repair it.
3: First, try to find a feasible replacement with minimum cost for the client in question with a client belonging to S_2 of the products already in the campaign. If there is such a replacement, perform it and search for another client otherwise continue trying to repair it.
4: Second, try to find a feasible replacement with minimum cost for the client in question with a client belonging to S_2 of the same product. If there is such a replacement, perform it and search for another client otherwise continue trying to repair it.
5: Third, if the number of clients in S_1 is greater than O_j remove the client, otherwise the product can not be included in the campaign, and finish the repair procedure for this product.
6: Update S_1 and S_2, if necessary. Continue the repairing process until no more clients are to be analyzed.
7: After checking or repairing all clients, if the product satisfies all problem constraints include it in the campaign, otherwise withdraw the product from the campaign. Finish the repair procedure for this product.

B. TABU SEARCH

The TS [31], [32] 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 [3] 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 $x'_{ij} = 0$ and $x'_{hj} = 1$ are set. The set $N_2(x, y)$ contains the feasible solutions (x', y') obtained from (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 $x'_{ij} = 0$, $x'_{hj} = 1$, $x'_{il} = 1$, and $x'_{hl} = 0$ are set. Examples of N_1 and N_2 can be seen in Figures 3 and 4.

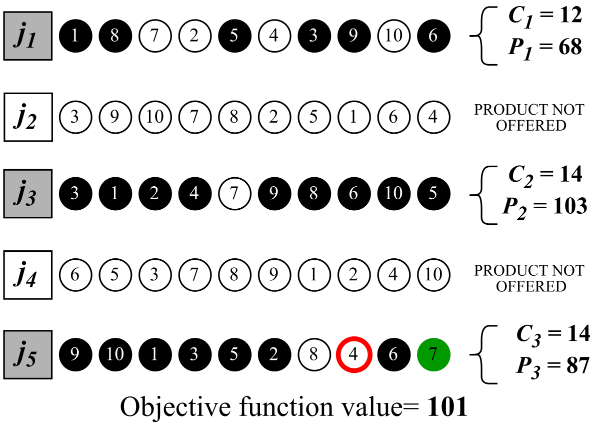


FIGURE 3. **Neighborhood N1:** $j = 5, i = 4, h = 7$
$x_{45} = 1$ *and* $x_{75} = 0 \rightarrow x_{45} = 0$ *and* $x_{75} = 1$.

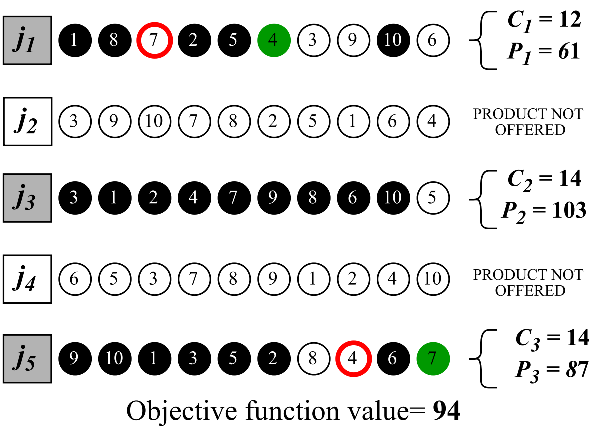


FIGURE 4. **Neighborhood N2:** $j = 5, l = 1, i = 4, h = 7$
$x_{45} = 1$ *and* $x_{75} = 0 \rightarrow x_{45} = 0$ *and* $x_{75} = 1$
$x_{41} = 0$ *and* $x_{71} = 1 \rightarrow x_{41} = 1$ *and* $x_{71} = 0$.

Neighborhood 3, $N_3(x, y)$, tries to combine the best features of $N_1(x, y)$ and $N_2(x, y)$, in a way that allows changes in the number of clients allocated to each active product (i.e., the cardinality of the set of clients allocated to each product), and at the end it inserts new offers into the final solution. The swap movement in $N_3(x, y)$ is represented by a tuple (j, l, i), meaning that the client i can be moved from product j to l. Initially, each client will be sorted in a non-increasing order according to their profit variance. Profitable single swaps between active products will be discovered and up to two profitable infeasible swaps will be stored. Each time a swap is found, a search for an interchange movement with another client that makes the whole operation feasible will be performed. After all clients are examined, there is a final trial to insert profitable offers in the active products. One example can be seen in Figure 5. This neighborhood effectively changes the number of clients offered by each product. This feature combined with the GA that changes the products that have been offered in the campaign, enables GATeS to address all classes of the proposed instances as shown in the experiments later on.

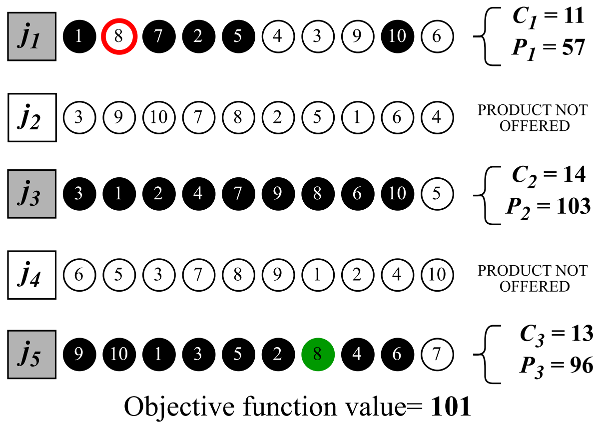


FIGURE 5. **Neighborhood N3:** $j = 1, l = 5, i = 8, (1, 5, 8)$**, i.e., the client 8 assigned to the offer 1 goes to the offer 5.**

The TS algorithm proposed here traverses the neighborhoods overcoming the convergence rate issues found in [3], mainly in large instances, where each movement has a small effect on the objective function. The first improvement is executing any profitable movement as soon as it is discovered. Additionally, the size of neighborhoods for N_1 and N_2 is managed as follows: at the beginning, only 70% of the clients sorted in non-increasing order of their NPP_{ij} are considered for analysis, and at every 10 iterations without improvement, this percentage grows 10% (up to 100%) to allow more combinations. The percentage is reset to the initial value every time an improvement solution is found. Notice that the concept of using different neighborhoods is proposed in the Variable Neighborhood Search (VNS) algorithm [27]. These modifications speed up TS and do not affect the solution quality.

The tabu list in the TS algorithm is updated in the following way: anytime a non-improvement movement is performed the reversal movement is forbidden only for 2 iterations without improvement. Remember that each iteration can involve multiple moves, as it performs all the improvement moves of N_1, N_2, and N_3, and all of them are included in the tabu list. This policy seeks a balance between the solution quality and the computational effort. TS performs 8 runs with different initial solutions and each run performs 30 iterations without improvement or 300 seconds. The way TS is inserted in GATeS can be seen in Figure 8. Note that, TS manages all GATeS hybridization with GA as a long-term memory that seeks to improve the solutions of its population as well as maintain its diversity.

In addition, GA is used as a pool of possible alternative ways to continue the search when TS cannot find improvement moves for two consecutive iterations. TS will request the best solution (not explored by TS yet) in the current GA population, having a fitness score of at most 3% away from the current objective function value. At each iteration, without improvement, this percentage is increased by 3%. At each iteration, without improvement, this percentage is increased by 3% to allow more elite solutions to be considered as the current solution in TS. When a new improvement solution is found or a new population is generated this percentage value returns to 3%.

C. GENETIC ALGORITHM

The GA [33] starts with a population of 100 individuals generated as described in section IV-A4. After that GA will optimize its individuals and start to be a long-term memory for TS, being called GA_Regression and GA_Optimization inside GATeS (see Figure 8. In the GA_Regression the current TS solution replaces the worst individual in the GA population and crossover and mutation are applied to build a new generation. In the GA_Optimization the entire population (as described in section IV-A4) is replaced, but that includes the best individual of the previous population before evolving it. Every call from GA produces 4 generations to allow the development of the population and to lose diversity. Each generation uses crossover and mutation to replace the entire population, except the best individual.

After initializing the population, GA starts the reproduction process, generating a new one that replaces entirely the old population but keeping the best-evaluated individual (i.e., elitism) in the next generation. Each new individual (i.e. offspring) is created by the crossover (see Section IV-C2 and Figure 7) between two selected parents from the previous population. The parents are selected through a weighted roulette wheel based on fitness (i.e., the objective function value). Then, two types of mutation procedures are applied to the new individuals. The first mutation is designed for instances considering cannibalism and it will be applied if the offspring is worse than the best solution so far (called Elite), changing the dominant parent with the lowest fitness parent, and applying again the crossover procedure. This will change the order of blocking cannibal products, producing different offspring.

The second mutation comes from the fact that TS interacts with the GA at each execution loop as shown in Figure 8. Section II shows that: a) TS does not use any frequency measurement as long-term memory, and b) the GA population quickly converges to solutions close to the best TS solution. Considering these observations, a diversification strategy is proposed where the GA population works as long-term memory since it keeps the information about the trajectory within the search space. This strategy is applied to 75% of the offspring worse than the Elite and tries to improve diversity offering a pool of solutions that highlights more frequent characteristics. Frequency analysis in the current population is applied by creating a list of the most frequent offers with a size of at most 10% of the number of clients. Is included in this list every client that takes part in at least 55% of the individuals in the current population. The mutation tries to change the value of up to 20% of the most frequent offers keeping the offspring feasible. These numbers are chosen to make a small change in the individuals using information about the population, noting that only 2% of the individuals are considered candidates for change. The list of most frequent offers is updated at the end of each generation. At that point, it can be observed that GA was not built to be only efficient but to create a pool of solutions the most diverse possible, creating a long-term memory for TS, and making GATeS more powerful.

1) SOLUTION REPRESENTATION

To enable the GA to better perform the crossover operation, and quickly converge on good solutions, cluster analysis (based on NPP_ij or cost) in correlated groups of possible clients to be offered is made instead of analyzing clients individually. Due to this reason, a different structure for the chromosome is proposed. Initially, the individual representing a solution is designed to have each client represented by a gene, where each gene (one for each product possible to be offered for the client) carries an additional set of information about that client. Each individual contains information including solution fitness, profit, cost, remaining budget for each product, and a Boolean vector indicating which products are being offered in the solution. The examples use this structure to represent the client status in each product offered. All this information contributes to speeding up the generation of new individuals. The chromosome is built as a linked list of genes as the structure shown in Figure 6. Observe that this chromosome resembles more an agent from memetic algorithms [34] that can carry more information than only the genetic structure.

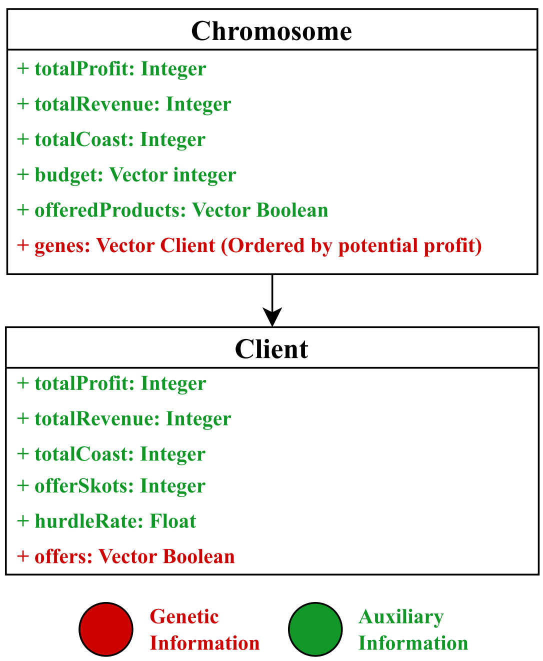


FIGURE 6. Structure of the representation of individuals for GA.

2) CROSSOVER

The crossover operation tries to transfer to offspring the best client offers among its selected parents, that is, the offspring is generated by choosing for each client belonging to *Parent1* or *Parent2*, the most profitable one. After that, the offspring must be checked and repaired if any problem constraint is violated. Considering that the parents carry different product combinations, in order to repair the offspring, first try to include clients (in non-increasing order of NPP_{ij}) in the products that still do not reach the minimal number of clients. If some product cannot be repaired or if it is the less profitable of a cannibal pair, it must then be excluded from the offspring. Notice how powerful is this operation since the offspring can have a completely different product combination than their parents, that characteristic does not appear in any other operator used by TS.

Once a feasible offspring is found, GA tries to maximize its profits by adding feasible offers to the solution. This is made in two steps: the genetic transfer, and the inclusion of new offers. For the genetic transfer, each gene that does not take part in the initial crossover will be checked and all feasible offers that can be made will be included in the offspring. Ultimately, for each product that still has an available budget, the algorithm tries to include clients in a non-increasing order of NPP_{ij}. A representation of the crossover procedure can be seen in Figure 7, where Parent1 (at the top) is identified as dominant since it has the higher fitness, the clients assigned to the offers have the circles filled in red, Parent2 (at the bottom) have the clients assigned to the offers with circles filled in blue. The offspring (in the middle) has the clients assigned to the offers filled with the same color as their original parents. The offspring also has a grey-filled circle that represents a genetic transfer from Parent 2, and a yellow-filled circle that represents an offer completion.

Since crossover compares parents' genes in a given order, the first genes will have a higher impact on the solution than the later ones. For this reason, the order to explore the genes is based on the customer's expected profit, i.e., the sum of all non-negative expected profits for each client, and then they are explored in a non-increasing order of that value. This is done to improve the resulting offspring quality.

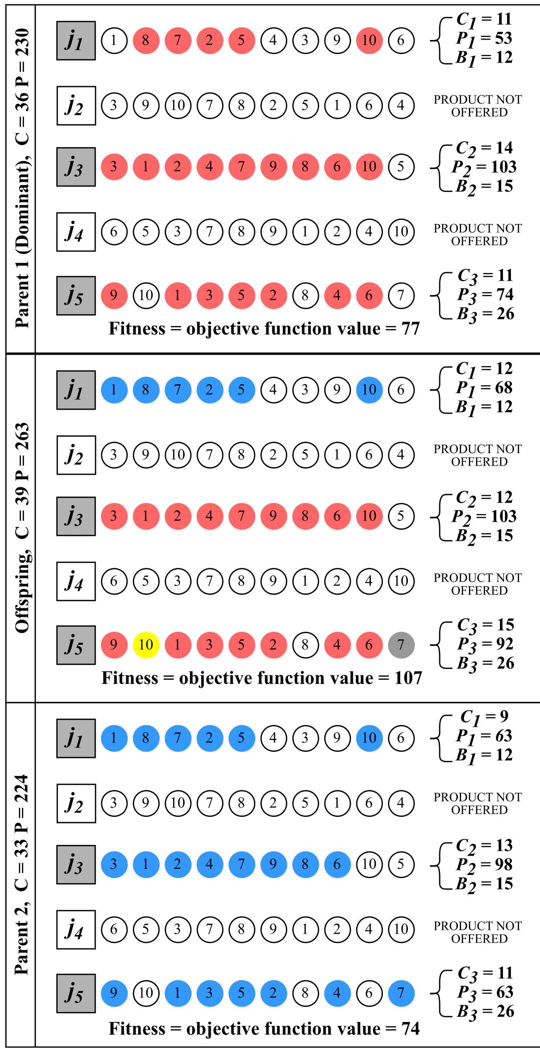


FIGURE 7. Crossover operator of GA for the direct marketing problem.

V. EXPERIMENTAL SETUP

After a long study about DMP, GATeS has been designed. To show the quality of results and the robustness of the proposed method, GATeS is compared with the two heuristic approaches proposed by [9], the TS extracted from GATeS, and the exact solutions (or upper bounds) obtained by commercial solvers (GUROBI and CPLEX), for the whole set of instances including or not the cannibalism constraint. GATeS is also compared against the results obtained from the literature by [3] (TS) and [17] (GRASP/VNS), but only for a small group of instances and not considering cannibalism. The initialization of GATeS and the number of runs performed for each instance were explained in Subsection IV-A4. The test instances used in the experiments are obtained from [3] 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, are used, resulting in 324 instances called Group 1.

To further investigate the robustness of GATeS, two sets of very large instances are generated, the first, named XL, follows the generation method of [3] in which all test instances use 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) resulting in 16 instances called Group 2. The second set of very large instances, named EST, maintain the intermediate random value of B_j and the tighter M_i as in [3] with m = 5000; n = 40; R = 15% (1 instance); and m = 15000; n = 10;

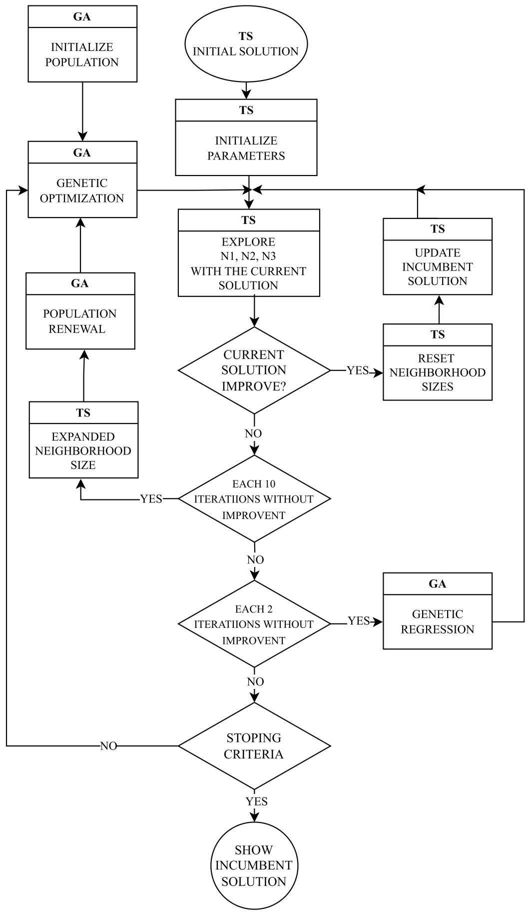


FIGURE 8. Overall flowchart of GATeS for the direct marketing problem.

$R = 10\%$ (16 instances); and $m = 100000$; $n = 15$; $R = 15\%$ (1 instance); resulting in 18 test instances called Group 3. Each instance (in all groups) has a cannibal version where a set with $(n/5)$ cannibal pair of products for each mode (Euclidean distance, Similarity, and Dissimilarity) defined in Section III-D is generated.

Wilcoxon sum rank tests are used for statistical analysis [35]. The effect size that indicates the practical significance of a research outcome is analyzed [36]. In particular, a large effect size means that a research finding has practical significance, while a small effect size indicates limited practical applications. The computer used to carry out the experiments has a Ryzen9 5900x processor, 32 GB 3200 MHz DDR4 RAM, RTX 3080 10 GB graphics card, and Windows 10 operating system. Table 4 summarizes the software resources to implement and execute the algorithms.

A. PROBLEM INSTANCE GENERATOR

Clients' databases may contain different consumption behaviors but also different purchase power that would generate significantly different consumption patterns. There are also differences among the products, e.g., similar products can have differences in quality and price when reaching different market segments. Hence, the m clients are randomly split into three Stratus: Stratus 1 - high purchase power, corresponding to a uniformly distributed random number in the interval $[0.06, 0.15] \times m$; Stratus 2 - intermediate purchase power, corresponding to $[0.15, 0.30] \times m$; Stratus 3 - low purchase power, corresponding to the remaining m. The maximum number of offers M_i that a client can receive is directly proportional to its purchase power, then M_i is a random number generated in the integer intervals [4, 6], [2, 4], and [1, 2] for Stratus 1, 2, and 3, respectively.

Products are divided into 4 types with different combinations of penetration range (i.e., the percentage of clients that will respond positively to the product) among each client stratus, the minimum and maximum turnover of positive offer responses, and the cost of the product offer for each client stratus. Each product will have the following attributes: product penetrations for each client stratus, the range of turnover for positive responses to the product offer, the cost of a single product offer for each client stratus, fixed cost for including the product on the set of offered products, product budget for offers, and minimum offer quantity requirement. These attributes can be summarized in Tables 5 and 6, where $[a, b]$ is a uniformly distributed random number generated between a and b, P_j^s is the penetration value of product j for client stratus S, T_j and T_j^M are, respectively, the minimum and maximum turnover values of the positive response to offers for the product j, c_j^S is the cost of offer product j to stratus S, MC_j is the minimum number of clients at product j, ts is the turnover seed for the products (defined by the user to generate diversity between instances) and cs is the cost seed for the products (defined by the user to generate diversity between instances).

To define which clients will be selected as positive responses to the product j, the number of clients belonging to a given client stratus will be the ones that respond positively to the product offer given by the formula: $C_j^s = P_j^S C^s$, where C^S is the client number in stratus S. For each selected client in the set PC_j^S, an individual offer will be randomly chosen in the interval $[T_j, T_j^M]$. After defining the number of offers to the clients in PC_j^S, the budget and the fixed cost are

TABLE 4. Summary of implementation resources.

Algorithm	References	Computational Resources
Exact	[22] and [25]	IBM ILOG CPLEX Optimization Studio [37] and ZIMPL [38]
HR-1	[9] with heuristic rule 1	Gurobi 8.1.1 [39] with Julia Pro 1.2.0-1 [40] and JuMP [41]
HR-2	[9] with heuristic rule 2	Gurobi 8.1.1 [39] with Julia Pro 1.2.0-1 [40] and JuMP [41]
Tabu	Present Paper	Visual Studio Community using C++ [42]
GATeS	Present Paper	Visual Studio Community using C++ [42]

TABLE 5. Parameters for the instance generator - Part 1.

	Product - Type 1			Product - Type 2		
Client Stratus	1	2	3	1	2	3
P_j^S	$[80, 100]\%$	$[15, 25]\%$	0%	$[60, 95]\%$	$[65, 95]\%$	$[5, 10]\%$
c_j^S	$6cs$	$[2cs, 4cs]$	$(c_j^1 + c_j^2)/2$	$4cs$	$[cs, 2cs]$	$(c_j^1 + c_j^2)/2$
T_j	$P_j^S(6ts + 10tsH)$			$P_j^S(3ts + 7tsH)$		
T_j^M	$[16ts, 24ts]$			$[8ts, 12ts]$		
MC_j	$[0.01m, 0.04m]$			$[0.05m, 0.15m]$		

TABLE 6. Parameters for the instance generator - Part 2.

	Product - Type 3			Product - Type 4		
Client Stratus	1	2	3	1	2	3
P_j^S	$[30, 40]\%$	$[75, 95]\%$	$[40, 60]\%$	0%	$[75, 90]\%$	$[85, 100]\%$
c_j^S	$3cs$	$[1.5cs, 2cs]$	$(c_j^1 + c_j^2)/2$	$3cs$	$[0.5cs, cs]$	$(c_j^1 + c_j^2)/2$
T_j	$P_j^S(2ts + 3tsH)$			$P_j^S(ts + tsH)$		
T_j^M	$[3ts, 6ts]$			$[2ts, 3ts]$		
MC_j	$[0.15m, 0.35m]$			$[0.15m, 0.55m]$		

defined. The budget for each product must meet at least 40% of customers who would respond positively to the offer, and it is calculated as: $B_j = \sum_{i \in PC_p^j}(T_{ij}/C_j^{S_i})$, where: T_{ij} is the turnover of product offer j to the client i and $C_j^{S_i}$ represents the cost of offering the product j to the client i belonging to the stratus S_i. The fixed cost F_j is randomly generated in the interval $[1.6B_j, 3.0B_j]$.

The feasibility of each instance is validated and if an instance contains products whose set of offers cannot reach the hurdle rate, they undergo a repair process where each offer will have a value correction applied to its turnover respecting the upper bounds. The turnover value of the most profitable offers is adjusted so that the client set matches the minimum offer requirement to achieve the hurdle rate. In total 1432 instances are generated as follows:

- 1296 instances of Group 1, 324 instances belong to the Original Problem (OP) without the cannibalism constraint, and 324 for each type of cannibalism presented, i.e., Euclidean Distances (ED), Similarity (Sim) and Dissimilarity (Diss);
- 64 instances of Group 2, following the same logic of Group 1, namely, 16 OP, 16 ED, 16 Sim, and 16 Diss;
- 72 instances of Group 3, i.e., 18 of each OP, ED, Sim, and Diss.

In the experiments, the upper bound (UB) is the optimum value (or the best bound found by the solver), and the lower bound (LB) is the objective function value obtained by the heuristic methods. The metric used to compare and evaluate the quality of the methods is the percentage gap defined as $\Delta = [(UB - LB)/UB]100\%$. To be fair, the instances that could not be solved by HR-1 or HR2 are not been considered in the statistics.

VI. RESULTS AND DISCUSSION

In this section GATeS is compared with two other algorithms from the literature GRASP/VNS proposed by [17] and Tabu Search [3] (called here TabuN to differ from our TS) for a selected number of test instances belonging to Group1 without cannibalism (see Section VI-1). Notice that in the original papers, no execution time was presented. This approach is not used for further analysis because the source codes were not available. In fact, both approaches are outperformed by the matheuristic proposed in [9].

TABLE 7. Comparison of GATeS, TabuN [3], and GRASP/VNS [17] for a selected set of Group1 test instances without cannibalism.

Instances	Average UB	Average Gaps ($\Delta\%$)					
				GRASP/VNS			
		GATeS	TabuN	$\gamma = 0.0$	$\gamma = 0.2$	$\gamma = 0.6$	$\gamma = 0.8$
S3-5	2432.11	**1.52**	6.86	10.68	6.90	6.79	**6.77**
S3-10	4703.44	**1.79**	6.52	6.98	7.14	**6.45**	6.50
S3-15	7018.28	**2.44**	7.76	7.77	7.92	**7.58**	7.49
M1-5	8345.94	**0.52**	**7.22**	9.72	**7.22**	8.47	7.24
M1-10	15751.39	**2.39**	8.54	9.12	8.67	9.10	**8.42**
M1-15	27710.06	**3.37**	**7.60**	12.15	7.69	7.89	7.63
M2-5	16546.67	**1.60**	9.75	11.29	10.26	**9.68**	9.85
M2-10	34389.44	**2.74**	**9.58**	9.90	9.59	9.76	9.78
M2-15	48767.11	**3.52**	**9.11**	11.15	9.34	9.91	9.44
L-5	85414.61	**1.83**	**10.86**	12.99	11.24	**10.86**	11.51
L-10	160995.44	**3.03**	**11.04**	12.29	10.94	11.22	11.93
L-15	244123.83	**4.63**	10.23	12.15	10.74	10.11	**10.22**
Mean	54016.53	**2.45**	**8.76**	10.52	8.97	8.99	8.90

Following the initial experimental setup, GATeS is further compared to TS (extracted from GATeS), HR-1, and HR-2 (see Section VI-2) using the instances without cannibalism (called original problem) and with Cannibalism (generated by Euclidean distance, Similarity, and Dissimilarity). Note that HR-1 and HR-2 failed to solve some instances from Group 1 (see Table 8), so it was decided to check if it was an isolated behavior or if it was a real lack of robustness. In the present investigation is also been presented a comprehensive statistical analysis including effect size (see Table 11) to demonstrate the superiority of GATeS.

Finally, the results obtained by all instances of Group 3 are analyzed. These instances are closer to the real world since millions of clients are considered (see Section VI-3). Again, it can be observed that HR-1 and HR-2 are not able to solve many of these instances, and GATeS outperforms the competing methods in solution quality while being competitive in computation time.

1) COMPARISONS OF GATES AGAINST GRASP/VNS AND TABUN

Table 7 presents the results for the instances solved by GRASP/VNS proposed by [17] and TabuN [3] for some instances belonging to Group1 (these instances appear in both original papers) without cannibalism. The GRASP/VNS results consist of different values of γ, which is the parameter used to randomize the constructive algorithm proposed by [3]. If $\gamma = 1$ the selection is completely greedy (original algorithm based on NPP_{ij} value), whereas if $\gamma = 0$ is completely random. Once more, it can be observed that GATeS outperforms the GRASP/VNS variants and TabuN in all test cases.

In fact, for the entire set of tested instances, GATeS presents a mean gap of 2.45% this is at least 5% larger than the other methods showing a very promising way to solve DMP. The numbers in bold in the last five columns indicate the best performance between the TabuN [3] and GRASP/VNS [17]. From these results, it can be noted that even the best between both methods perform worse than GATeS.

2) COMPARISONS OF GATES AGAINST TABU, HR-1 AND HR-2

This Section starts by analyzing the robustness of HR-1 and HR-2 for the whole set of instances, that means, Group 1, 2, and 3 with and without cannibalism, and observe if HR-1 and HR-2 can solve them. Table 8 shows that there are still unsolved problems in Group 1 and Group 3 either with or without cannibalism. It can be also observed that for Group 3 (larger and more difficult to solve instances) HR-2 presents much more weakness in solving only half of them, and for the first time, HR-1 and HR-2 have unsolved instances for the same group despite there being no instances in common. Even though HR-1 and HR2 are much faster than the heuristics the fact that they have unsolved instances is not practical. This fact already justifies the use of robust heuristics such as GATeS.

TABLE 8. **Number of unsolved instances by HR-1 and HR-2.**

Heuristic Algorithm	Number of Unsolved Instances							
	Original Problem		Cannibalism					
			Euclidean Distance		Similarity		Dissimilarity	
	Group1	Group3	Group1	Group3	Group1	Group3	Group1	Group3
HR–1	3	1	3	1	3	1	3	1
HR–2	0	9	0	9	0	9	0	9

The results including the heuristic methods developed in this research (say a Tabu Search and GATeS) compared against HR-1 and HR-2 are presented in Table 9 and the TS is included in the comparisons to prove the efficiency of hybridization with GA since it is the base of GATeS. This can be considered as an ablation test. Table 9 shows the number of the best solutions found by each algorithm in each group of instances with and without cannibalism. The table also shows when the methods find the same solution. From Table 9, it can be observed that GATeS, HR-1, and HR-2 are competitive in instances that belong to Groups 1 and 2, but in Group 3 the superiority of GATeS is remarkable recalling that GATeS solves all instances. Another thing to observe is that HR-1 and HR-2 find many identical solutions while GATeS does not show that the diversity of solutions found by GATeS together with their quality makes it more robust.

Once Table 9 is analyzed, it is possible to realize the general performance of the methods looking at Table 10 which shows the percentage gaps of the algorithms for Groups 1, 2, and 3 with and without cannibalism.

From Table 10 it can be observed that GATeS outperforms Tabu, HR-1, and HR-2 in all test cases. In fact, GATeS outperforms HR-1 and HR-2 even when the best between the two competing algorithms is considered (last column of Table 10). The only two exceptions are for Group 1 (Original Problem and Cannibalism Dissimilarity). Note that in both cases there are instances that are not solved. In the next section, it will be shown that there are instances with gaps greater than 50% in all algorithms making a statistical analysis very important to check whether the results are significant or not and if the effect size is large.

TABLE 9. **Number of times in which best solutions are obtained by GATeS, Tabu, HR-1, and HR-2 for Group 1,2, and 3 test instances.**

	Number of Best Results Found											
				Cannibalism								
	Original Problem			Euclidean Distance			Similarity			Dissimilarity		
Group	1	2	3	1	2	3	1	2	3	1	2	3
GATeS	84	1	11	97	8	5	112	4	3	76	3	6
Tabu	23	0	0	21	0	1	3	0	2	6	0	1
HR-1	112	2	4	82	2	5	86	9	3	82	11	2
HR-2	38	13	0	47	3	0	61	0	0	87	1	0
GATeS=Tabu	1	0	2	0	0	5	0	0	7	0	0	6
GATeS=HR-1	2	0	0	4	0	0	2	0	0	2	0	0
GATeS=HR-2	3	0	0	3	0	1	3	0	0	8	0	0
Tabu=HR-1	1	0	0	1	0	0	2	0	0	2	0	0
Tabu=HR-2	1	0	0	2	0	0	0	0	1	0	0	0
HR-1=HR-2	48	0	0	54	3	0	43	3	0	49	1	0
GATeS=Tabu=HR-1	0	0	0	0	0	1	1	0	1	1	0	1
GATeS=Tabu=HR-2	1	0	1	1	0	0	1	0	1	0	0	2
GATeS=HR-1=HR-2	8	0	0	4	0	0	9	0	0	10	0	0
Tabu=HR-1=HR-2	2	0	0	5	0	0	0	0	0	1	0	0
GATeS=Tabu=HR-1=HR-2	0	0	0	3	0	0	1	0	0	0	0	0
Total instances	324	16	18	324	16	18	324	16	18	324	16	18

Although the performance of HR-1 and HR-2 is competitive with GATeS in Group1 test instances, their performance is significantly degraded in Group2 and Group3 test instances. In fact, some test instances are not solved by HR-1 and HR-2 as shown in Table 8. HR-1 does not resolve three instances of Group 1 and one instance of Group 3, whereas HR-2 does not resolve half of the instances (9) of Group 3. Even if the unresolved instances are not the same there is no guarantee that this will occur every time. In contrast, GATeS is able to resolve all instances of all three groups and shows the best average performance. These results demonstrate not only the good performance of GATeS in terms of solution quality but also its robustness in any type of test instance. As mentioned before, the results of GATeS and Tabu are the best of 8 runs with different initial solutions. The time shown in Table 8 represents the total time spent for the eight runs. Although the time that was spent for GATeS is high, it can be observed that the gap is much smaller for instances closer to the real world. Therefore, a tenth of percentages can represent a lot of monetary resources making this effort worthwhile.

After observing the good performance of GATeS looking only for the results, statistical analysis to validate the results obtained is required. Wilcoxon sum rank tests are used to compare the algorithms [35]. Table 11 presents the statistical analysis consisting of the comparisons of GATeS vs HR-1, GATeS vs HR-2, HR-1 vs HR-2, GATeS vs Tabu, Tabu vs HR-1, and Tabu vs HR-2. Each comparison determines if there is a significant difference between the compared algorithms and quantifies the magnitude of the differences between the related samples. The effect size measure (named r) is also performed [36], [43].

TABLE 10. **Average percentage gaps of GATeS, HR-1, and HR-2 for Groups 1,2 and 3 test instances.**

Group	Problem	GATeS		Tabu		H-R1		H-R2		Average between best Δ% H-R1 and H-R2
		Δ%	Time (s)	Δ%	Time (s)	Δ%	Time (s)	Δ%	Time (s)	
1	Original Problem	2.55	346.87	6.56	916.48	5.82	1.55	4.35	1.19	**2.12**
	Cannibalism-Euclidean Distance	**2.07**	340.57	6.71	945.94	6.33	1.33	5.13	1.19	2.72
	Cannibalism-Similarity	**2.24**	339.37	11.69	1020.89	5.93	1.37	5.31	1.28	2.95
	Cannibalism-Dissimilarity	2.94	333.43	11.41	825.57	6.29	1.44	3.71	1.30	**2.08**
2	Original Problem	**4.30**	12 870.38	17.16	4771.19	7.91	80.56	6.81	132.78	6.59
	Cannibalism-Euclidean Distance	**5.03**	9100.19	22.67	4617.13	15.41	36.23	15.60	37.12	14.90
	Cannibalism-Similarity	**8.60**	7975.19	25.28	4445.75	14.54	41.13	19.60	77.87	14.52
	Cannibalism-Dissimilarity	**6.82**	7684.94	21.69	4230.88	13.13	42.14	16.83	77.09	12.80
3	Original Problem	**5.34**	3385.44	19.37	2226.00	16.81	15.14	42.53	8.71	14.69
	Cannibalism-Euclidean Distance	**8.61**	3323.67	16.47	1451.28	28.07	10.27	39.83	9.32	21.76
	Cannibalism-Similarity	**3.70**	3017.06	10.71	1799.33	23.33	15.12	36.69	11.81	17.88
	Cannibalism-Dissimilarity	**6.13**	3388.72	12.88	2112.17	25.10	15.03	37.20	11.17	19.84

Obs.: the unsolved cases were not considered or penalized.

TABLE 11. **Statistical summary of comparisons between algorithms.**

Statistical Summary												
Algorithm 1 vs Algorithm 2	Group 1				Group 2				Group 3			
	OP	Cannibalism			OP	Cannibalism			OP	Cannibalism		
		ED	Sim	Dissim		ED	Sim	Dissim		ED	Sim	Dissim
GATeS vs HR-1	++(~)	++(*)	++(*)	++(*)	++(~)	++(~)	++(~)	++(~)	++(*)	++(*)	++(*)	++(*)
GATeS vs HR-2	+(*)	++(*)	++(*)	+(*)	++(*)	++(~)	++(~)	++(~)	++(*)	++(*)	++(*)	++(*)
HR-1 vs HR-2	-(*)	-(*)	-(*)	--(~)	-(*)	=(~)	++(*)	++(*)	++(*)	++(~)	++(~)	++(~)
GATeS vs Tabu	++(*)	++(*)	++(*)	++(*)	++(*)	++(*)	++(*)	++(*)	++(*)	++(~)	++(~)	++(~)
Tabu vs HR-1	-(*)	=(*)	--(*)	--(*)	--(*)	--(~)	--(~)	--(~)	--(~)	++(~)	++(*)	++(~)
Tabu vs HR-2	--(*)	-(*)	--(*)	--(*)	--(*)	--(~)	--(~)	--(~)	++(*)	++(~)	++(*)	++(~)
Per Group GATeS vs HR-1	++(*)				++(*)				++(*)			
Per Group GATeS vs HR-2	++(*)				++(*)				++(*)			
Per Group HR-1 vs HR-2	-(*)				+(*)				++(~)			
Per Group GATeS vs Tabu	++(*)				++(*)				++(*)			
Per Group Tabu vs HR-1	--(*)				--(*)				++(*)			
Per Group Tabu vs HR-2	--(*)				--(*)				++(*)			
All instances GATeS vs HR-1	++(*)											
All instances GATeS vs HR-2	++(*)											
All instances HR-1 vs HR-2	-(*)											
All instances GATeS vs Tabu	++(*)											
All instances Tabu vs HR-1	--(*)											
All instances Tabu vs HR-2	--(*)											

Effect size: Large | Moderate | Small | Insignificant

A total of 96 groups are been compared, as shown in Table 11, where the legend representing the results is the following:

- (*): The two samples are significantly different, they have statistically significant differences;
- (∼): The two samples are similar, they have no statistically significant differences;
- ++: Algorithm 1 has a average gap ≥ 2% compared to Algorithm 2;
- +: Algorithm 1 has an average ≥ 0.5% compared to Algorithm 2;
- - -: Algorithm 2 has a average ≥ 2% compared to Algorithm 1;
- -: Algorithm 2 has an average ≥ 0.5% compared to Algorithm 1;
- =: Algorithm 1 has a performance similar to Algorithm 2 (i.e., average gap ≤ 0.5%)

The cells have been colored to represent the magnitude of the effect size of the comparisons [36], [43]:

- Green - Large ($r \geq 0.5\%$): considers a high-impact effect in the comparisons. The association is marked and has a considerable impact from a practical point of view.
- Yellow - Moderate ($r \geq 0.3\%$): indicates a substantial effect on comparisons. The association is noticeable and may have significant practical relevance.
- Red - Small ($r \geq 0.1\%$): indicates modest strength of the difference between the results of the two related

samples. The association is slight and may not be as substantial from a practical point of view.
- Gray - Insignificant ($r < 0.1$).

The first six rows of Table 11 analyze the instances by group and type of problem, the next six rows analyze the instances gathered by group, and in the last six rows, the whole set of instances is analyzed. The symbol ($\sim$) indicates that there is no significant difference between the compared algorithms. For all cells in Table 11, involving GATeS there are none with no significant difference and gap < 0.5, and in all of them, GATeS presents better behavior in terms of gap (+ or ++). When each group and each instance type are considered separately, there are 72 possible combinations, and the concentration of samples where there is no significant difference are the ones involving HR-1 and HR-2 as we expected, but there are many occurrences involving Tabu and HR-1, and Tabu and HR-2, justifying again the use of GATeS to solve the problem. When each group is considered individually (six lines in the middle of the table), only in one case there is no significant difference and involves the pair HR-1 and HR-2 as expected.

Lastly, when the entire set of instances is considered, there is no case presenting significant differences. For all these results and from the effect size analysis comes to the conclusion that the generated instances are suitable for algorithmic comparisons and have practical applications. Consequently, the conclusion is that the investigated algorithms are significantly different and their performance can be compared by the percentage deviation from the optimal.

Better performance is been expected from HR-1 and HR-2 since they are matheuristics. However, when considering the whole set of solved instances the comparisons show 410 cases with an advantage for GATeS, 57 for Tabu, 400 for HR-1, 250 for HR-2, and 315 ties (63.8% of them were between HR-1 and HR-2, but only in 4 cases all methods found the same solution), as shown in Table 9.

Analyzing the mean and median values of the percentage gaps, the research will have the best results the lower the mean also there will be good results if the median is close to the mean. It is important to observe that the statistics indicate that GATeS has an advantage over HR-1, with mean and median values of 2.81 and 1.30, and 7.22 and 1.75, respectively. When comparing GATeS against HR-2 it can be observed again that GATeS has an advantage, with mean and median values of 2.67 and 1.28, and 5.97 and 2.65, respectively. Note that, the mean and median values are slightly different when comparing GATeS with HR-1 and HR-2 because only the pairs of instances solved by both methods to compute these values are considered, remembering that GATeS has solved all instances. Ultimately, when comparing HR-1 against HR-2, there are 630 cases with an advantage for HR-1, 444 for HR-2, and 306 ties, with mean and median values of 6.92 and 1.66, and 6.02 and 2.68, respectively, demonstrating a similar behavior with a little advantage to HR-2.

The mean and median values when comparing GATeS against Tabu are 2.79 and 9.94 for the mean, and 1.27 and 7.48 for the median, respectively which shows a significant advantage to GATeS. In fact, this demonstrates the effect of hybridization between TS and GA. When Tabu is compared against HR-1, Tabu has 10.03 and 7.63 for the mean and median, respectively, whereas the values for HR-1 remain the same demonstrating an advantage in favor of HR-1. A similar observation occurs when Tabu is compared against HR-2 with Tabu 9.93 and 7.63 for the mean and median, respectively, whereas the values for HR-2 remain the same. These comparisons show that the performance of GATeS in all cases can be directly related to the hybridization proposed in this paper.

Looking again at Table 11, the lines where GATeS is compared against HR-1 and HR-2 in instances of Group 3 (larger and more difficult to solve) besides the significant difference observed, all the effect sizes are Moderate or Large, indicating the practical application of these instances. Once again the superiority of GATeS is confirmed.

Concluding this statistical analysis the set of instances is significantly different. Instances of Group 3 are the ones closest to practical applications in which GATeS outperforms the competing methods. For this reason, the discussion in the next section focuses on instances belonging to Group 3.

3) COMPARISONS OF GATES IN INSTANCES FROM GROUP 3

Table 12 presents the results for all instances of Group3 for the exact method, GATeS, Tabu, HR-1, and HR-2. The name of instances follows one code, e.g., EST-5K-15-40-8-3-10-15-40-35, is an instance of Group 3 (EST), having $m = 5000$, $H = 15\%$, $n = 40$, $ts = 8$, and $cs = 3$, the percentage of products type 1, 2, 3, and 4 are 10%, 15%, 40%, and 35%, respectively. The instances of Group 3 are described in Section V-A and they are instances considered big and challenging to solve. This is because their solution space is challenging and has a small number of optimal solutions to be explored. These test instances were selected because they are closer to reality as shown by the effect size analysis. Note that HR-1 and HR-2 cannot find solutions for the entire set of instances, and when they do, they have enormous variability this demonstrates the weakness and lack of robustness of those matheuristics applied to this kind of instances.

It can be also observed in Table 12, that GATeS outperforms Tabu, HR-1, and HR-2, in almost all Group 3 instances, regarding the percentage deviation from the optimum (or the upper bound found by the exact solver). This performance is more significant when the instances include the cannibalism constraint. In addition, the number of unsolved instances by HR-1 and HR-2 is very high, corresponding to one and nine (half of 18) respectively. This fact demonstrates that the cannibalism constraint makes the problem harder to solve and the instances belonging to Group 3 indeed test the performance of the considered methods. However, the performance of GATeS is still superior to the competing algorithms.

TABLE 12. Comparison of average percentage gaps between GATeS, Tabu, HR-1, and HR-2 for Group3 test instances.

Problem	Instances	Exact method	GATeS		Tabu		H-R1		H-R2	
			Δ%	Time (s)	Δ%	Time (s)	Δ%	Time (s)	Δ%	Time (s)
Original Problem	EST-100K-15-15-8-3-10-15-40-35	3 255 529.00	2.81	38 366.00	99.94	0.00	6.43	74.62	71.55	34.56
	EST-15K-10-10-10-10-10-15-40-35	29 214.00	2.90	1128.00	4.68	346.00	0.00	10.82	NS	NS
	EST-15K-10-10-10-11-10-15-40-35	84 283.00	5.03	839.00	7.48	4337.00	34.89	8.99	NS	NS
	EST-15K-10-10-10-12-10-15-40-35	47 441.00	0.00	826.00	0.00	570.00	27.33	8.96	NS	NS
	EST-15K-10-10-10-8-10-15-40-35	53 601.00	0.02	1521.00	0.03	50.00	1.07	15.93	NS	NS
	EST-15K-10-10-10-9-10-15-40-35	73 440.00	0.00	1162.00	42.33	541.00	25.89	11.52	NS	NS
	EST-15K-10-10-4-3-10-15-40-35	76 789.00	13.91	1662.00	26.40	5014.00	0.00	21.87	56.40	6.87
	EST-15K-10-10-4-4-10-15-40-35	42 210.00	3.24	1554.00	36.32	4311.00	17.83	7.86	NS	NS
	EST-15K-10-10-5-5-10-15-40-35	61 412.00	2.35	1487.00	3.52	4479.00	0.00	10.17	59.20	8.30
	EST-15K-10-10-6-5-10-15-40-35	91 334.00	4.94	1388.00	5.61	4495.00	53.21	17.73	52.98	5.30
	EST-15K-10-10-7-5-10-15-40-35	101 694.00	5.56	1914.00	5.58	587.00	11.32	15.54	NS	NS
	EST-15K-10-10-7-6-10-15-40-35	74 077.00	6.47	1833.00	11.92	371.00	36.83	12.86	15.73	5.68
	EST-15K-10-10-7-7-10-15-40-35	18 575.00	0.00	854.00	0.00	534.00	3.29	10.49	19.83	6.77
	EST-15K-10-10-8-3-10-15-40-35	6620.00	0.00	879.00	0.00	168.00	NS	NS	0.00	4.73
	EST-15K-10-10-8-7-10-15-40-35	62 535.00	6.00	1031.00	33.98	4195.00	32.10	11.13	NS	NS
	EST-15K-10-10-8-8-10-15-40-35	69 676.00	36.30	1874.00	24.48	4672.00	13.11	6.31	NS	NS
	EST-15K-10-10-9-10-10-15-40-35	48 842.00	3.31	1785.00	38.81	959.00	14.35	8.81	48.26	4.84
	EST-5K-15-40-8-3-10-15-40-35	183 906.00	3.23	835.00	7.61	4439.00	8.03	3.70	58.85	1.38
	Average	243 399.00	5.34	3385.44	19.37	2226.00	16.81	15.14	42.53	8.71
Cannibalism Euclidean Distance	EST-100K-15-15-8-3-10-15-40-35	2 563 025.00	7.62	35 522.00	99.93	0.00	90.33	57.14	63.86	47.31
	EST-15K-10-10-10-10-10-15-40-35	29 212.00	2.90	349.00	4.68	313.00	0.00	8.66	NS	NS
	EST-15K-10-10-10-11-10-15-40-35	65 954.00	56.99	644.00	0.00	433.00	30.88	8.63	NS	NS
	EST-15K-10-10-10-12-10-15-40-35	47 441.00	0.00	1668.00	0.00	560.00	50.48	5.30	NS	NS
	EST-15K-10-10-10-8-10-15-40-35	53 601.00	0.01	1695.00	0.02	419.00	27.10	9.10	NS	NS
	EST-15K-10-10-10-9-10-15-40-35	73 440.00	0.00	1653.00	42.33	554.00	25.89	10.06	NS	NS
	EST-15K-10-10-4-3-10-15-40-35	76 789.00	13.91	2520.00	19.60	4661.00	7.92	6.20	56.40	7.54
	EST-15K-10-10-4-4-10-15-40-35	42 209.00	3.23	1700.00	3.23	377.00	17.83	7.20	NS	NS
	EST-15K-10-10-5-5-10-15-40-35	61 412.00	2.35	1737.00	3.52	4623.00	0.00	5.77	59.20	7.05
	EST-15K-10-10-6-5-10-15-40-35	60 903.00	8.73	1675.00	11.20	5033.00	45.43	9.83	29.48	3.14
	EST-15K-10-10-7-5-10-15-40-35	101 689.00	5.55	1738.00	10.13	630.00	0.00	8.68	NS	NS
	EST-15K-10-10-7-6-10-15-40-35	62 424.00	0.00	1337.00	51.15	1392.00	42.95	6.71	0.00	3.58
	EST-15K-10-10-7-7-10-15-40-35	18 575.00	0.00	1513.00	0.00	494.00	3.29	10.78	19.83	8.58
	EST-15K-10-10-8-3-10-15-40-35	6620.00	0.00	402.00	0.00	178.00	NS	NS	0.00	2.33
	EST-15K-10-10-8-7-10-15-40-35	49 486.00	0.01	1444.00	0.01	306.00	26.09	7.76	NS	NS
	EST-15K-10-10-8-8-10-15-40-35	69 676.00	36.30	1668.00	24.48	4641.00	13.11	5.77	NS	NS
	EST-15K-10-10-9-10-10-15-40-35	48 842.00	13.28	1716.00	13.28	533.00	14.35	6.73	48.25	4.06
	EST-5K-15-40-8-3-10-15-40-35	176 762.00	4.08	845.00	12.89	976.00	81.48	0.20	81.47	0.27
	Average	200 448.00	8.61	3323.67	16.47	1451.28	28.07	10.27	39.83	9.32
Cannibalism Similarity	EST-100K-15-15-8-3-10-15-40-35	2 407 391.00	0.48	30 168.00	99.93	0.00	90.30	47.64	61.53	39.74
	EST-15K-10-10-10-10-10-15-40-35	27 847.00	0.00	232.00	0.00	336.00	0.00	10.71	NS	NS
	EST-15K-10-10-10-11-10-15-40-35	84 283.00	21.75	985.00	7.48	4337.00	34.89	8.99	NS	NS
	EST-15K-10-10-10-12-10-15-40-35	47 441.00	0.00	1874.00	0.00	570.00	27.33	8.96	NS	NS
	EST-15K-10-10-10-8-10-15-40-35	27 905.00	0.02	951.00	0.02	38.00	2.05	16.81	NS	NS
	EST-15K-10-10-10-9-10-15-40-35	73 440.00	0.00	1496.00	42.33	541.00	25.89	10.06	NS	NS
	EST-15K-10-10-4-3-10-15-40-35	67 222.00	4.67	2148.00	8.47	4564.00	2.65	11.09	50.19	7.00
	EST-15K-10-10-4-4-10-15-40-35	40 844.00	0.00	1459.00	0.00	672.00	16.35	15.56	NS	NS
	EST-15K-10-10-5-5-10-15-40-35	40 984.00	0.00	1429.00	0.00	3475.00	0.00	21.08	51.23	18.37
	EST-15K-10-10-6-5-10-15-40-35	87 958.00	1.12	1913.00	1.99	4464.00	54.07	18.01	51.17	12.66
	EST-15K-10-10-7-5-10-15-40-35	83 507.00	5.21	1627.00	5.21	3132.00	17.11	9.71	NS	NS
	EST-15K-10-10-7-6-10-15-40-35	62 424.00	0.00	1807.00	0.00	888.00	42.95	21.20	0.00	8.63
	EST-15K-10-10-7-7-10-15-40-35	18 575.00	0.00	1473.00	0.00	534.00	3.29	10.49	19.83	8.58
	EST-15K-10-10-8-3-10-15-40-35	6620.00	0.06	1039.00	0.00	168.00	NS	NS	0.00	2.33
	EST-15K-10-10-8-7-10-15-40-35	49 486.00	0.01	1204.00	0.01	330.00	26.09	18.18	NS	NS
	EST-15K-10-10-8-8-10-15-40-35	58 770.00	26.27	1739.00	10.46	4636.00	19.13	13.39	NS	NS
	EST-15K-10-10-9-10-10-15-40-35	42 356.00	0.00	1824.00	0.00	539.00	34.26	7.99	40.33	7.80
	EST-5K-15-40-8-3-10-15-40-35	171 662.00	6.97	939.00	16.95	3164.00	0.17	7.20	55.91	1.22
	Average	188 818.00	3.70	3017.06	10.71	1799.33	23.33	15.12	36.69	11.81
Cannibalism Dissimilarity	EST-100K-15-15-8-3-10-15-40-35	2 776 856.00	15.70	33 152.00	99.95	0.00	90.22	49.82	66.64	33.54
	EST-15K-10-10-10-10-10-15-40-35	27 847.00	0.00	366.00	0.00	1310.00	0.00	9.49	NS	NS
	EST-15K-10-10-10-11-10-15-40-35	84 283.00	21.75	1665.00	7.48	4337.00	34.89	8.99	NS	NS
	EST-15K-10-10-10-12-10-15-40-35	47 441.00	0.00	1830.00	0.00	570.00	27.33	8.96	NS	NS
	EST-15K-10-10-10-8-10-15-40-35	27 905.00	0.02	1588.00	0.02	177.00	2.05	13.93	NS	NS
	EST-15K-10-10-10-9-10-15-40-35	73 440.00	0.00	1879.00	42.33	541.00	25.89	10.06	NS	NS
	EST-15K-10-10-4-3-10-15-40-35	64 263.00	17.11	3164.00	33.36	3777.00	26.12	21.12	47.90	12.50
	EST-15K-10-10-4-4-10-15-40-35	40 844.00	0.00	1684.00	0.00	3888.00	16.35	15.56	NS	NS
	EST-15K-10-10-5-5-10-15-40-35	40 984.00	0.00	1250.00	0.00	3806.00	0.00	14.10	51.23	13.91
	EST-15K-10-10-6-5-10-15-40-35	87 958.00	1.12	2265.00	1.99	4008.00	54.07	15.10	51.17	12.47
	EST-15K-10-10-7-5-10-15-40-35	96 067.00	0.03	2761.00	0.05	3590.00	13.56	14.41	NS	NS
	EST-15K-10-10-7-6-10-15-40-35	62 424.00	0.00	1695.00	0.00	1245.00	42.95	18.99	0.00	9.55
	EST-15K-10-10-7-7-10-15-40-35	18 575.00	0.00	1568.00	0.00	534.00	3.29	10.49	19.83	8.58
	EST-15K-10-10-8-3-10-15-40-35	6620.00	0.00	483.00	0.00	168.00	NS	NS	0.00	2.33
	EST-15K-10-10-8-7-10-15-40-35	49 486.00	0.01	1799.00	0.01	1488.00	26.09	18.07	NS	NS
	EST-15K-10-10-8-8-10-15-40-35	68 631.00	35.33	1873.00	25.91	4062.00	4.76	16.03	NS	NS
	EST-15K-10-10-9-10-10-15-40-35	42 356.00	0.00	1152.00	0.00	1264.00	34.26	7.89	40.33	6.52
	EST-5K-15-40-8-3-10-15-40-35	178 743.00	19.18	823.00	20.67	3254.00	24.79	2.46	57.66	1.15
	Average	210 818.00	6.13	3388.72	12.88	2112.17	25.10	15.03	37.20	11.17

Observing the first line of Table 12, where the instance EST-100K-15-15-8-3-10-15-40-35 is analyzed, GATeS have a percentual gap of 2.81 while Tabu, HR-1, and HR-2, 99.94 (in fact, Tabu is unable to solve this particular instance), 6.43, and 71.55, respectively. This behavior is worse when cannibalism is considered since HR-2 degenerates enormously. It can be also observed that the computational effort spent by GATeS is larger than HR-1 and HR-2. However, a campaign is planned for a period (i.e., monthly) and a difference that in some cases is close to 90% worth the effort.

VII. DATA AVAILABILITY STATEMENT

The data that supports the findings of this study is openly available on the GitHub repository at https://github.com/gp-direct-marketing/GATeS_Results, under the MIT license.

VIII. CONCLUSION

This paper has presented a hybrid algorithm, named GATeS, which can be briefly described as a multiple start TS with an embedded GA that provides the diversification necessary to better traverse the solution space, and as known until this point, it is the first time that a GA has been used as long-term memory in a TS procedure. This combination achieves a better exploitation and exploration of the search space. The computational results combined with the statistical analysis demonstrate that GATeS performs better than other state-of-the-art algorithms, in terms of solution quality and robustness. The consideration of the constraint on cannibalism to the DMP has challenged the competing algorithms, while GATeS was able to tackle it without sacrificing its performance. In fact, while the competing algorithms failed to solve some large-scale problem test instances, GATeS was able to solve them.

The set of benchmark instances used in the experiments demonstrate the weaknesses and strengths of the algorithms mainly when the cannibalism constraint is considered. GATeS outperforms the competing algorithm considering the three kinds of proposed instances - with and without cannibalism -, and this fact becomes more evident in realistic instances from Group 3 including cannibalism.

In conclusion, GATeS is a promising method to solve DMP, with and without the cannibalism constraint. Even though GATeS requires more computational effort, it still remains usable in the real world because the DMP is a long-term planning task. It is worth mentioning that all the problem instances, methods, and results (e.g., optimal solutions found by commercial solvers) are publicly available.

ACKNOWLEDGMENT

The authors are indebted to the three anonymous reviewers for their constructive suggestions that have helped to improve the original manuscript.

REFERENCES

[1] P. Kotler and G. M. Armstrong, *Marketing Mix: Selected Chapters From: Principles of Marketing*, P. Kotler G. Armstrong, Eds. London, U.K.: Pearson, 2018.

[2] V. S. Abedi, "Allocation of advertising budget between multiple channels to support sales in multiple markets," *J. Oper. Res. Soc.*, vol. 68, no. 2, pp. 134–146, Feb. 2017.

[3] F. T. Nobibon, R. Leus, and F. C. R. Spieksma, "Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms," *Eur. J. Oper. Res.*, vol. 210, no. 3, pp. 670–683, May 2011.

[4] P. Ladyzynski, K. Zbikowski, and P. Gawrysiak, "Direct marketing campaigns in retail banking with the use of deep learning and random forests," *Exp. Syst. Appl.*, vol. 134, pp. 28–35, Nov. 2019.

[5] T. Bigler, M. Kammermann, and P. Baumann, "A matheuristic for a customer assignment problem in direct marketing," *Eur. J. Oper. Res.*, vol. 304, no. 2, pp. 689–708, Jan. 2023.

[6] V. L. Miguéis, A. S. Camanho, and J. Borges, "Predicting direct marketing response in banking: Comparison of class imbalance methods," *Service Bus.*, vol. 11, no. 4, pp. 831–849, Dec. 2017.

[7] N. Van Praag, "Optimization promotion campaigns using Tabu search," Ph.D. thesis, Faculty Bus. Economics, KU Leuven, Leuven, Belgium, 2010.

[8] A. Czikosova, M. Antošova, and K. Cukolova, "Strategy in direct and interactive marketing communications," *Proc.-Social Behav. Sci.*, vol. 116, pp. 1615–1619, Jan. 2014.

[9] F. Cetin and C. Alabas-Uslu, "Heuristic solution to the product targeting problem based on mathematical programming," *Int. J. Prod. Res.*, vol. 55, no. 1, pp. 3–17, Jan. 2017.

[10] E. S. Oblander, S. Gupta, C. F. Mela, R. S. Winer, and D. R. Lehmann, "The past, present, and future of customer management," *Marketing Lett.*, vol. 31, no. 2, pp. 125–136, 2020.

[11] V. Nesterenko, R. Miskiewicz, and R. Abazov, "Marketing communications in the era of digital transformation," *Virtual Econ.*, vol. 6, pp. 57–70, Mar. 2023.

[12] J. Haupt and S. Lessmann, "Targeting customers under response-dependent costs," *Eur. J. Oper. Res.*, vol. 297, no. 1, pp. 369–379, Feb. 2022.

[13] F. Cui, H. Hu, and Y. Xie, "An intelligent optimization method of e-commerce product marketing," *Neural Comput. Appl.*, vol. 33, no. 9, pp. 4097–4110, May 2021.

[14] D. Pachamanova, V. S. Y. Lo, and N. Gülpinar, "Uncertainty representation and risk management for direct segmented marketing," *J. Marketing Manag.*, vol. 36, nos. 1–2, pp. 149–175, Jan. 2020.

[15] M.-D. Cohen, "Exploiting response models—Optimizing cross-sell and up-sell opportunities in banking," *Inf. Syst.*, vol. 29, no. 4, pp. 327–341, Jun. 2004.

[16] T. Bhaskar, R. Sundararajan, and P. G. Krishnan, "A fuzzy mathematical programming approach for cross-sell optimization in retail banking," *J. Oper. Res. Soc.*, vol. 60, no. 5, pp. 717–727, May 2009.

[17] T. A. Oliveira, V. N. Coelho, M. J. Souza, D. L. Boava, F. Boava, I. M. Coelho, and B. N. Coelho, "A hybrid variable neighborhood search algorithm for targeted offers in direct marketing," *Electron. Notes Discrete Math.*, vol. 47, pp. 205–212, Feb. 2015.

[18] V. B. de Souza, "Uma abordagem meta-heurística para o problema de marketing direto considerando o canibalismo entre produtos," M.S. thesis, PPGA—Federal Univ. Santa Maria, 2018.

[19] F. M. Müller, I. S. Bonilha, and V. B. Souza, "A hybrid approach (genetic algorithm + Tabu search) to solve the direct marketing problem," in *Proc. 19 Latin-Iberoamerican Conf. Oper. Res.*, 2018.

[20] M. B. Traylor, "Cannibalism in multibrand firms," *J. Consum. Marketing*, vol. 3, no. 2, pp. 69–75, Feb. 1986.

[21] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Series of Books in the Mathematical Sciences). W. H. Freeman, Ed., 1st ed. 1979.

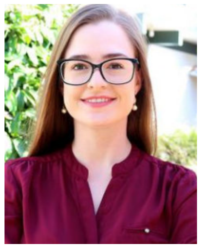
[22] V. A. Schneider, "Análise de modelos matemáticos para o problema de marketing direto com oferta de produtos," M.S. thesis, PPGEP—Federal Univ. Santa Maria, 2019.

[23] V. N. Coelho, T. A. Oliveira, I. M. Coelho, B. N. Coelho, P. J. Fleming, F. G. Guimarães, H. Ramalhinho, M. J. F. Souza, E.-G. Talbi, and T. Lust, "Generic Pareto local search metaheuristic for optimization of targeted offers in a bi-objective direct marketing campaign," *Comput. Oper. Res.*, vol. 78, pp. 578–587, Feb. 2017.

[24] F. M. Müller, I. S. Bonilha, V. A. Schenider, V. B. Souza, G. R. Cruz, and M. Mavrovouniotis. (2023). *GATeS*. [Online]. Available: https://github.com/GATeS-DMKT/Final_Results

[25] G. R. Cruz, ''Solução do problema de marketing direto com canibalismo: Métodos exatos e heurísticos,'' M.S. thesis, PPGCC—Federal Univ. of Santa Maria, 2020.
[26] T. A. Feo and M. G. Resende, ''Greedy randomized adaptive search procedures,'' *J. Global Optim.*, vol. 6, no. 2, pp. 109–133, 1995.
[27] P. Hansen and N. Mladenović, ''Variable neighborhood search: Principles and applications,'' *Eur. J. Oper. Res.*, vol. 130, no. 3, pp. 449–467, May 2001.
[28] A. Kassambara, *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. Multivariate Analysis, 2017.
[29] H. Clifford and W. Stephenson, *An Introduction to Numerical Classification*. Cambridge, MA, USA: Academic Press, 1975.
[30] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W. H. Freeman, 1973.
[31] F. Glover, ''Tabu search—Part I,'' *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, 1989.
[32] F. Glover, ''Tabu search—Part II,'' *ORSA J. Comput.*, vol. 2, no. 1, pp. 4–32, Feb. 1990.
[33] J. H. Holland, ''Genetic algorithms,'' *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.
[34] P. Moscato, ''On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms,'' California Inst. Technol., Pasadena, CA, USA, Tech. Rep., 826, 1989.
[35] F. Wilcoxon, ''Individual comparisons of grouped data by ranking methods,'' *J. Econ. Entomology*, vol. 39, no. 2, pp. 269–270, Apr. 1946.
[36] C. O. Fritz, P. E. Morris, and J. J. Richler, ''Effect size estimates: Current use, calculations, and interpretation,'' *J. Experim. Psychol., Gen.*, vol. 141, no. 1, pp. 2–18, 2012.
[37] *IBM ILOG CPLEX Optimization Studio—CPLEX User's Manual*, 2017.
[38] T. Koch, ''Rapid mathematical programming,'' Technische Univ. Berlin, Germany, Tech. Rep., ZIBTR 04–58, 2004.
[39] Gurobi, *Gurobi Optimizer Reference Manual*, 2020.
[40] JuliaPro, *Installation Manual and Quickstart for Linux*, 2021.
[41] I. Dunning, J. Huchette, and M. Lubin, ''JuMP: A modeling language for mathematical optimization,'' *SIAM Rev.*, vol. 59, no. 2, pp. 295–320, Jan. 2017.
[42] Visual Studio, *Visual Studio Manuals*, 2019.
[43] J. Cohen, *Statistical Power*, 2nd ed. Lawrence Erlbaum Associates, 1988.

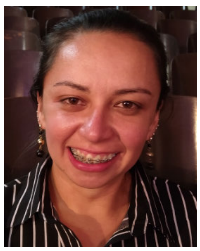
FELIPE MARTINS MÜLLER received the degree in electrical engineering from the Federal University of Santa Maria, Santa Maria, Rio Grande do Sul, Brazil, in 1987, and the M.Sc. and Ph.D. degrees in electrical engineering (automation) from UNICAMP, Campinas, São Paulo, Brazil, in 1990 and 1993, respectively. From September 1991 to September 1992, he was a Visiting Researcher with the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Montreal, Canada. From August 2014 to August 2015, he was a Visiting Professor with the Centre for Computational Intelligence, De Montfort University, Leicester, U.K. From 2009 to 2013, he was the Rector of the Federal University of Santa Maria, where he is currently a Titular Professor with the Applied Computing Department. He is the coauthor of more than 35 publications. He has published in *International Transactions in Operational Research*, *Production Planning and Control*, *Transportation Science*, *Computing and Operations Research*, *International Journal of Production Economics*, IEEE Transactions on Cybernetics, and several international conferences. His research interests include heuristics, metaheuristics, evolutionary algorithms, dynamic optimization problems, and their applications. From 2017 to 2022, he was the President of the Operational Research Brazilian Society (SOBRAPO), where he is currently a member.



VANESSA A. SCHNEIDER was born in Trs de Maio, Rio Grande do Sul, Brazil, in 1991. She received the degree in production engineering from Faculty Trs de Maio—SETREM, in 2017, the degree from the Special Graduate Program of Teacher Training for Professional Education, Federal University of Santa Maria, in 2019, and the master's degree in production engineering from the Graduate Program in Production Engineering, in 2019. She is currently pursuing the Ph.D. degree in production engineering with the Federal University of Santa Maria. Her research interest includes quantitative methods for decision-making.

IAÊ SANTOS BONILHA received the degree in computer science from the Federal University of Santa Maria, in 2012, and the master's degree in computer science from the Graduate Program in Computer Science, Federal University of Santa Maria, in 2014, with a focus on real-time systems. He is currently pursuing the Ph.D. degree in production engineering with the Federal University of Santa Maria. He is a Data Scientist with Progress Rail, Brazil.

VERIDIANE BARBARA DE SOUZA received the degree in business from Faculdade Integrada de Santa Maria (FISMA), in 2016, and the master's degree in business from the Graduate Program in Business, Universidade Federal de Santa Maria, in 2018, with a focus on marketing information systems. She is currently pursuing the Ph.D. degree in production engineering with Universidade Federal de Santa Maria.



GREICI DA ROSA DA CRUZ received the degree in information systems from Universidade Franciscana, in 2015, the Specialist degree in database management from the Lutheran University of Brazil, in 2018, the degree from the Special Graduate Program of Teacher Training for Professional Education, Federal University of Santa Maria, in 2018, and the master's degree in computer science from the Graduate Program in Computer Science, Federal University of Santa Maria, in 2020. His research interest includes applied computing.

MICHALIS MAVROVOUNIOTIS received the B.Sc. degree in computer science from the University of Leicester, U.K., in 2008, the M.Sc. degree in natural computation from the University of Birmingham, U.K., in 2009, and the Ph.D. degree in computer science from the University of Leicester, in 2013. He is currently a Researcher with the ERATOSTHENES Centre of Excellence, Limassol, Cyprus. He has more than 60 refereed publications. His research interests include evolutionary computation, swarm intelligence, memetic computing, combinatorial optimization problems, artificial intelligence in dynamic and uncertain environments, and relevant real-world applications. He is the Chair of the IEEE Task Force on Evolutionary Computation in Dynamic and Uncertain Environments through the Evolutionary Computation Technical Committee of the IEEE Computational Intelligence Society.