

HEURISTIC ALGORITHM FOR UNIVARIATE STRATIFICATION PROBLEM

JOSÉ ANDRÉ BRITO^{1,*}, GUSTAVO SEMAAN², LEONARDO DE LIMA³
AND AUGUSTO FADEL⁴

Abstract. In sampling theory, stratification corresponds to a technique used in surveys, which allows segmenting a population into homogeneous subpopulations (strata) to produce statistics with a higher level of precision. In particular, this article proposes a heuristic to solve the univariate stratification problem – widely studied in the literature. One of its versions sets the number of strata and the precision level and seeks to determine the limits that define such strata to minimize the sample size allocated to the strata. A heuristic-based on a stochastic optimization method and an exact optimization method was developed to achieve this goal. The performance of this heuristic was evaluated through computational experiments, considering its application in various populations used in other works in the literature, based on 20 scenarios that combine different numbers of strata and levels of precision. From the analysis of the obtained results, it is possible to verify that the heuristic had a performance superior to four algorithms in the literature in more than 94% of the cases, particularly concerning the known algorithm of Lavallée–Hidiroglou.

Mathematics Subject Classification. 90C59, 62D05.

Received March 13, 2023. Accepted September 26, 2023.

1. INTRODUCTION

Official statistical institutes make available, with great frequency, a wide range of data and statistics of great relevance to governments, with regard, for example, to the planning and implementation of their public policies, as well as to the citizen willing to accompany them. For the most part, these data are produced from surveys based on probabilistic sampling [40].

According to Lohr [40] and Cochran [14], sampling techniques allow the collection and production of statistics of interest in less time, with lower cost and greater accuracy. However, when are considered administrative issues, and there is a requirement of the desired minimum precision about the statistics produced, it is necessary to incorporate stratification in the planning/design phase of the sample in search of the least possible variability. Stratified sampling is well used together with other techniques such as simple random sampling and cluster sampling [40].

Keywords. Stratification, minimum sample, allocation, optimization, algorithms.

¹ National School of Statistical Sciences, Rio de Janeiro, Brazil.

² Fluminense Federal University, Rio de Janeiro, Brazil.

³ Federal University of Paraná, Curitiba, Brazil.

⁴ Brazilian Institute of Geography and Statistics, Rio de Janeiro, Brazil.

*Corresponding author: jambrito@gmail.com

When solving the stratification problem (univariate in this article), the N observations of an X variable (stratification variable), known for the entire population, are distributed among L population strata by determining $L - 1$ cutoff points, called strata boundaries. To determine such cutoff points, one of the following objectives functions should be considered: (i) setting a sample of size n , minimizing a variance expression (or the coefficient of variation) associated with variable X , or (ii) setting a level of precision (the target coefficient of variation) associated with variable X , minimize the total sample size n that will be allocated to L strata.

Determining the cutoff points considering one of these objectives (versions) is equivalent to solving a nonlinear integer optimization problem that is difficult to solve computationally. Therefore, in recent decades, several methods have been proposed, basically classified as approximation or optimization [29], and most of them has considered objective (i).

This article considers a hybrid approach that works to identify the cutoff points of the strata through a stochastic optimization method and then, with the obtained cutoff points, the exact method proposed by Brito *et al.* [9], perform the optimal allocation of the sample to the strata. More specifically, it is proposed a heuristic algorithm based on the Biased Random Key Genetic Algorithms (BRKGA) meta-heuristic [25], which we call STRATMH, for the univariate stratification problem using objective function (ii) as described above. It is essential to underline that the proposed heuristic differs from the classical BRKGA, particularly concerning the forms of representation and construction of the non-codified solutions, dispensing with solutions (de)coding procedures. More specifically, each chromosome produced in the initial generation and from the application of the crossover and mutation operators corresponds to the solutions of the stratification problem: the strata sizes. Another differential concern is the crossover operator, proposed specifically for the problem and which has the effect of a local search.

Adopting such a method in this approach constitutes a differential from the other approaches proposed in the literature for the stratification problem, in which the sample sizes allocated to the strata are calculated using classic allocation methods from the literature, which do not consider a criterion of optimization. As a result, such methods generally produce non-integer sample sizes, which are rounded *a posteriori*, producing sample sizes corresponding to a local optimum or even an infeasible solution (which do not meet the restriction of the level of precision – coefficient of variation). The exact approach guarantees the global optimum for the allocation problem and the lowest cost for carrying out the survey sampling since there is a direct relationship between the number of units that must be interviewed in the sample and such cost.

The article is organized as follows: Section 2 presents the univariate stratification problem, considering these two possible objectives, as well as a literature review regarding the main methods applied to its resolution. Section 3 brings a discretization proposal for this problem and the proposed heuristic. Finally, Section 4 presents the results and analyzes considering a set of 31 populations and 20 different scenarios in relation to the number of strata and levels of precision.

2. UNIVARIATE STRATIFICATION PROBLEM

Consider a population P with N units defined from a set $U = \{1, 2, \dots, N\}$ and the stratification variable X . Then, to obtain a good estimate t_x for the variable X and a set of Y variables investigated in the survey, the population of total size N is divided into L subpopulations, with N_h units (where $h = 1, \dots, L$ and $N_1 + \dots + N_L = N$) called population strata [40] and denoted by $U_1, U_2, \dots, U_L$.

To carry out this division, the population vector $X_U = (x_1, x_2, \dots, x_N)$ with N values of X associated to the population units is defined such that $x_1 \leq x_2 \leq \dots \leq x_N$. Population strata are determined using $(L - 1)$ cutoff points $b_1 < \dots < b_h < \dots < b_{L-1}$ such that $b_h \in [x_1, x_N]$ for each $h = 1, \dots, L - 1$, and according to (1)–(3):

$$U_1 = \{i \in U : x_i \leq b_1\} \quad (1)$$

$$U_h = \{i \in U : b_{h-1} < x_i \leq b_h\}, \quad h = 2, 3, \dots, L - 1 \quad (2)$$

$$U_L = \{i \in U : b_{L-1} < x_i\}. \quad (3)$$

From the determination of the U_h ($h = 1, \dots, L$) strata it is immediately observed that $|U_1| + \dots + |U_L| = N$ and the values of N_h and S_{xh}^2 , corresponding respectively to population size and population variance of X in stratum h , are obtained. In a later step, considering the fixed sample size (n), as stated in Problem (I), or a level of precision, defined in terms of the target coefficient of variation (cv_t) associated with the estimator of the variable X , as stated in Problem (II), the allocation [14, 40] of a sample of size n_h ($h = 1, \dots, L$) is done. Therefore, n_h is the number of sample units allocated to the stratum h , out of the total population in the stratum (N_h). See Appendix C (p. 3058) for a comprehensive list of the notation adopted.

When solving the univariate stratification problem, one must determine the cutoff points $b_1, b_2, \dots, b_{L-1}$ and the values of n_h (where $n_1 + \dots + n_L = n$ and $2 \leq n_h \leq N_h \forall h = 1, \dots, L$), which correspond to the sample allocation, that is, the size of the sample to be independently selected in each stratum, to obtain the minimum variance (or the coefficient of variation) given that the sample size n is fixed, which corresponds to solving Problem I; or obtain the minimum sample size for a fixed level of precision cv_t , which corresponds to solving Problem II. Problems I and II are defined as follows:

Problem I

$$\text{Minimize } V(t_x) = \sum_{h=1}^L N_h^2 \frac{S_{xh}^2}{n_h} \left(1 - \frac{n_h}{N_h}\right) \quad (4)$$

Subject to:

$$n_1 + \dots + n_L = n \quad (5)$$

$$2 \leq n_h \leq N_h, \quad h = 1, \dots, L \quad (6)$$

$$n_h \in Z_+, \quad h = 1, \dots, L. \quad (7)$$

Problem II

$$\text{Minimize } n = \sum_{h=1}^L n_h \quad (8)$$

Subject to:

$$cv(t_x) \leq cv_t \quad (9)$$

$$2 \leq n_h \leq N_h, \quad h = 1, \dots, L \quad (10)$$

$$n_h \in Z_+, \quad h = 1, \dots, L. \quad (11)$$

It is necessary to determine the cutoff points and the sample sizes n_h (decision variables) that minimize the respective objective functions and meet the constraints of these problems to obtain the optimum concerning Problem (I) or (II). The nonlinearity of Problem (I) is present in the objective function in (4), and the nonlinearity of Problem (II) is associated with the restriction of the precision level in (9), which guarantees that the resulting coefficient of variation of the total estimator ($cv(t_x)$) is equal to or less than the target coefficient of variation (cv_t).

According to several approaches in the literature proposed for solving (I) and (II), after determining the strata, it is common to apply the Neyman allocation (12) [40] to determine the sample sizes of the Problem (I). The equation (13) used in Problem (II) is obtained from the substitution of (12) in the term on the left side of equation (9) and algebraic manipulations.

$$n_h = n \frac{S_{xh} N_h}{\sum_{g=1}^L S_{xg} N_g}, \quad h = 1, \dots, L \quad (12)$$

$$n = \frac{\left(\sum_{h=1}^L N_h S_{xh}\right)^2}{cv_t^2 T_x^2 + \sum_{h=1}^L N_h S_{xh}^2}. \quad (13)$$

When using this allocation or other allocation methods in the literature [4, 14], non-integer sample sizes n_h , in general, are obtained. It is necessary to apply a rounding procedure to ensure that the integrality constraint is satisfied. The issue of doing so, is that such rounding can produce sample sizes that do not meet constraints (5) and (6) or (9) and (10), which are mandatory.

In order to address this issue, the heuristic proposed in Section 3 for solving the Problem (II) considers, in the phase of allocation of sample sizes to strata, the application of the exact method proposed by Brito *et al.* [9]. This method guarantees the overall optimal concerning sample allocation. More specifically, values $n_h \in Z_+(h = 1, \dots, L)$ such that the objective function of (II) has a minimum value and are fulfilled the constraints (9)–(11). Adopting this method in the second stage of the heuristic guarantees that the smallest possible sample size allocated to the strata for each stratification obtained is the global optimum in the allocation stage. A differentiating point concerning the various approaches proposed in the literature, which use classic allocation methods in this stage, which do not consider the optimization criterion, producing non-integer sample sizes, which need to be rounded, losing the global optimum. Thus, the heuristic solutions meet the precision constraint and ensure lower research costs related to the number of units that must be interviewed in the sample.

2.1. Literature review

Due to the relevance of the univariate stratification problem and its computational complexity, several algorithms are proposed in the literature, classified as approximate or optimization, according to [29]. Approximate algorithms are simple to implement, require few computational operations, and are not focused on searching for optimal cutoff points. These points, together sample size, allow the minimization of the objective function of Problem (I) or (II), which implies low stratification efficiency.

On the other hand, algorithms based on optimization often make possible to determine solutions of superior quality if compared to those obtained from approximated methods, although it demands intensive computation. In [19], the authors present a review of several methods from 1950 to the present. Next, we present other references associated with the main stratification methods.

Regarding the class of algorithms that deal with Problem (I), in [16], there is an algorithm based on the application of the Dalenius-Hodges rule. In [28], considering an extension of Ekman's rule [22], the strata are determined so that the variance of the stratification variable is minimal, with pre-fixed n and L and application of the Neyman allocation. In [27], a simple and easy-to-implement algorithm is proposed, which uses the general term formula of a geometric progression to determine the strata limits for asymmetric populations.

In [49], an algorithm based on the modified Newton method is proposed, which determines the limits of the strata and produces a local optima about the variance expression, based on the Neyman allocation and the choice of a good starting point. In [37], the authors evaluate the algorithms proposed in [27, 35, 38]. Finally, in [32], under the hypothesis that X has a normal or triangular distribution and that the sampling is done with replacement, an algorithm is proposed that determines the cutoff points of the strata, using the Dynamic Programming technique. This algorithm can be applied to minimize a variance expression, considering the use of uniform, proportional, or Neyman allocation.

In [8], an exact algorithm based on concepts of graph theory and the minimization of the expression of variance and application of proportional allocation is proposed. In [36], a comparative study between the algorithm proposed in [35, 37] and the algorithm proposed in [30] is presented. In [29], the authors compare optimization methods and approximate methods, considering the stratification of asymmetric populations. Finally, in [33, 34], algorithms based on the Dynamic Programming technique are proposed and simultaneously perform the stratification and allocation of the sample.

In [42], under the hypothesis that the stratification variable has a Weibull distribution, the stratification problem is solved using the Dynamic Programming technique and Neyman allocation. In [30], a genetic algorithm

that considers the minimization of variance is proposed, based on four possible types of sample allocation. In [17, 18], an approach based on dynamic programming for the two-variable stratification problem uses proportional and Neyman allocations. Finally, in [20], the bivariate stratification problem is also solved *via* dynamic programming, evaluating different probability distribution functions.

In [7], an algorithm is proposed based on the ILS (Iterated Local Search) method and Neyman's allocation. In [10, 11], are proposed algorithms based, respectively, on the BRKGA (Biased Random Key Genetic Algorithm) and GRASP (Greedy Randomized Adaptive Search Procedure) methods, using the exact method proposed in Brito *et al.* [9] in the allocation phase.

About the set of works that address the Problem (II), in [38], an algorithm for stratification of asymmetric populations was proposed that uses the power allocation [4] to determine the sample size. In [45], is proposed a generalization of the algorithm presented in [38], which incorporates two models that account for the discrepancy between the stratification variable and a research study variable. In [35], an algorithm called Random Search is proposed, which uses optimization concepts to determine the cutoff points of the strata and the Neyman allocation. In [1] several methods based on Kozak's [35] and Sethi's [47] algorithms are compared, concluding that the former, with initial strata of equal size, is more reliable and efficient than other approaches taken into account, and [2] introduces the R package *stratification* that implements methods used in [1], in addition to allowing to consider stratum-specific anticipated non-response and a model for the relationship between stratification and survey variables.

In [39], an algorithm is proposed that solves the multivariate stratification problem and uses a penalized objective function, which is optimized by applying the Simulated Annealing method. In [12], the authors propose a brute force algorithm, limited to small populations (N value), and an algorithm based on the VNS (Variable Neighborhood Search) method, where, in the sample allocation step, both do use of the exact method proposed by Brito *et al.* [9]. As a result of this method, the authors developed the R package *stratvns*. In [3], a genetic algorithm is proposed to solve the multivariate stratification problem, with possible application to the univariate case, which originated the R package *SamplingStrata*. In [43], the *stratifyR* package is designed by implementing a dynamic programming algorithm proposed by Khan *et al.* [31–33] to solve the optimal strata boundaries and the optimum sample sizes for the univariate problem. The *stratification* package [2] implements a generalized Lavallée-Hidioglou method [38] of strata construction using the Kozak or Sethi algorithm.

Combining hybrid metaheuristics with other heuristics or exact methods has become a widely used approach in optimization problems. A comprehensive survey on the topic is presented in [5]. More specifically, the strategy of using exact methods to improve genetic algorithms has proven successful in several applications, such as in [6], where a branch and bound procedure is probabilistically integrated into a genetic algorithm in order to enhance solutions obtained for a variant of the permutation flow shop scheduling problem. A hybrid metaheuristic combining genetic algorithm and linear programming to address the simultaneous order acceptance and scheduling problem is proposed in [44], achieving near-optimal solutions. In [15], multi-objective genetic algorithms are combined with iterated linear programming to improve computational efficiency by dividing the search space into sub-spaces for discrete and continuous variables, demonstrated through an example of optimal control valve placement in water distribution networks. In [26], a BRKGA is combined with a novel placement strategy and linear programming to optimize the unequal area facility layout problem and achieves improved solutions compared to other benchmark approaches. The time-dependent vehicle routing and scheduling problem with CO₂ emissions optimization is addressed in [50], combining genetic algorithm and dynamic programming to improve the efficiency and effectiveness of vehicle scheduling plans with lower CO₂ emissions. Finally, in [21], a linear programming-assisted genetic algorithm is applied to solving the flexible job shop lot streaming problem, where the algorithm searches over discrete and continuous variables and uses linear programming to refine promising solutions periodically, resulting in superior performance compared to embedded and parallel genetic algorithms.

It is worth mentioning that we have used the R packages: *stratification* [2], *stratvns* [12], and *SamplingStrata* [3], in the numerical experiments section, and their results compared to the BRKGA metaheuristic.

3. PROBLEM DISCRETIZATION AND HEURISTIC

3.1. Discretization

Before detailing the proposed heuristic, the discretization also considered in [8, 10] will be presented and used as a basis for representing the solutions produced by the heuristic. The discretization consists of considering only feasible candidates to cutoff points the ones available in X_U without repetition.

As defined in Section 2, the problem addressed in this article, in its original form, corresponds to an integer nonlinear optimization problem with a linear objective function and a nonlinear constraint associated with the coefficient of variation, which is a function of the allocated sample sizes to the (integer) strata and the cut points used to define the L strata. Initially, without considering the discretization of the problem, any real values in the interval $[\min(X_U), \max(X_U)]$ are, theoretically, candidates for the $L - 1$ cutoff points.

That is, in principle, there are infinite values in this interval that can be used for population stratification. However, depending on the distribution of the variable X , many subsets of real values of X , chosen as cutoff points, can produce strata with only one observation or even empty ones, considering equations (1)–(3) (Sect. 2).

For example, considering the variable $X_U = \{1, 2, 3, 3, 4, 5, 7, 8, 8, 9, 10, 12, 12, 15\}$ and $L = 3$, we have the interval $[1, 15]$. If, for example, $b_1 = 1.5$ and $b_2 = 2.1$ are taken, and equations (1)–(3) are applied, strata 1 and 2 would have only one observation each, leading to an infeasible solution.

Additionally, due to the way strata are defined, it is natural to use, as cutoff points, a subset of X_U values, a fact that allows determining, as explained below, the exact maximum number of solutions to the problem that must be evaluated and, in the case of enumerating all these solutions, to determine the optimal global solution, something that cannot be guaranteed, in principle, using any cutoffs points in $[\min(X_U), \max(X_U)]$. Another advantage of such discretization is ensuring the generated solutions' feasibility. Finally, discretization allows the application of approaches based on metaheuristics, notably known for their ability to produce reasonable quality solutions for combinatorial optimization problems with low computational cost.

From the definition of the stratification problem and the equations (1)–(3) presented in Section 2, it is possible, initially, to take the values of the vector X_U and gather it into a set B considering only the distinct values.

Then, from B , it is possible to determine a possible partition of X_U by the L population strata, more specifically, how many and which X_U values are in each stratum. Assumed the basic and necessary assumption (associated with constraints (6) and (10)) that $|B| \geq 2L$, guaranteeing at least one solution to the problem, this corresponds to determining a non-negative integer solution that satisfies constraints (14) and (15):

$$w_1 + \cdots + w_h + \cdots + w_L = |B| \quad (14)$$

$$w_h \geq 2, \quad h = 1, \dots, L. \quad (15)$$

In these constraints, each w_h corresponds to the number of observations of B that are in the h -th stratum. Constraint (14) guarantees that the sum of the total observations (N_h) in each of the strata corresponds to N , and constraint (15) implies that each stratum has at least two distinct observations.

From each solution $(w_1, \dots, w_L)$ that satisfies (14) and (15), the corresponding values of N_h and $S_{x_h}^2$ are obtained. Then, using equations (13) and (12), the values of n_h (already rounded) are determined. In the next step, the values of n_h , N_h , and $S_{x_h}^2$ are used to calculate the objective function of the Problem (II) and to assess whether constraints (9) and (10) are attended.

For example, assuming $L = 3$, $X_U = \{1, 2, 3, 3, 4, 5, 7, 8, 8, 9, 10, 12, 12, 15\}$ and using (14) and (15), are obtained $B = \{1, 2, 3, 4, 5, 7, 8, 9, 10, 12, 15\}$ and equations (16) and (17):

$$w_1 + w_2 + w_3 = 11 \quad (16)$$

$$w_1 \geq 2, w_2 \geq 2, w_3 \geq 2. \quad (17)$$

Analyzing (16) and (17), we have that $w = (4, 5, 2)$ is a possible solution. Since $w_1 = 4$, this implies that stratum 1 has the values 1, 2, 3, and 4, including the repeated values. Consequently, by the correspondence between B and X_U , one arrives at $N_1 = 5$ and, similarly, $w_2 = 5$ implies $N_2 = 6$ and $w_3 = 2$ implies $N_3 = 3$. This correspondence also makes it possible to calculate the $S_{x_h}^2$ variances.

From this discretization, solving the univariate stratification problem to obtain the optimum consists of determining, among all the possible solutions that satisfy (14) and (15), the one that simultaneously produces the cutoff points and the sample size that minimizes the objective function of the Problem (II). According to [12], the total of integer solutions of (14) and (15) is defined by:

$$\text{TS} = \frac{(|B| - L - 1)!}{(|B| - 2L)! (L - 1)!}. \quad (18)$$

Also, in [12], the authors proposed a brute force algorithm (BFA) that enumerates all solutions w that satisfies, simultaneously (14) and (15). For each solution w , the algorithm determines the associated values of N_h and $S_{x_h}^2$ ($h = 1, \dots, L$) and, in a later step, applies the method proposed in [9] to obtain the sample allocation. The solution w corresponding to the smallest sample size is the optimal solution to the Problem (II). However, according to [12], even considering all the computational apparatus currently available, the BFA has its application restricted to cases where $\text{TS} \leq 10^7$, a limit that is exceeded for small values, such as $N = 160$ and $L = 3$, for example. It is the main focus of the Computational Intractability theme, as exposed in [46].

Due to the complexity of the stratification problem and the impossibility of applying the AFB to stratify populations with a more significant number of observations, in this Section 3, we present a heuristic for the stratification problem corresponding to the optimization model defined in Problem (II). The proposed heuristic was based on discretization and on the application of BRKGA principles.

3.2. BRKGA

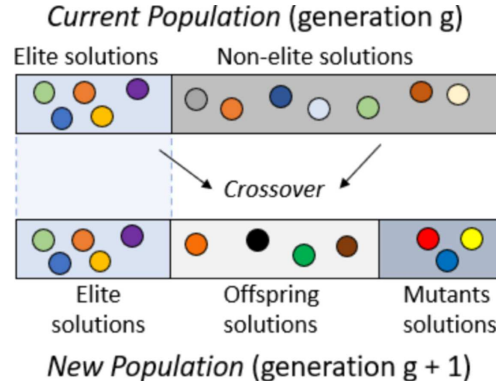
The biased random-key genetic algorithm (BRKGA) is a stochastic optimization method, also known as metaheuristics [25,41], that has been applied in several optimization problems, for example, in [23,24] and which has many similarities with genetic algorithms (GAs), which are a particular class of evolutionary algorithms [41].

In each generation (iteration) of the BRKGA, p solutions (vectors) associated with the optimization problem in question and corresponding to the BRKGA population are produced, evaluated (*via* objective function), and updated. Furthermore, to produce new solutions during BRKGA generations, ensuring diversity and quality, are applied procedures called selection, crossover, and mutation operators in each generation of this method.

To apply the selection and crossover operators, it is necessary to calculate the value of the objective function of the problem in question at each generation for each p solution (current population), ordering them increasingly (minimization problem) according to the objective function value. Then, considering such ordering, the best p_e solutions are allocated to a set of elite solutions denoted by S_e . Finally, the other solutions ($p - p_e$) are allocated to a set of non-elite solutions S_{ne} .

The selection operator is applied to ensure that the best solutions (*i.e.*, solutions with smaller objective function values) will be copied in the population for all generations. More specifically, are copied to the population p_e vectors of S_e , considered in the next generation. Finally, the other solutions for this new population are obtained from the application of the crossover and mutation operators.

The p_m solutions produced from mutation are also copied to the population evaluated in the following generation. These solutions are generated analogously to the first generation of BRKGA. Finally, to complement the population evaluated in the next generation, $(p - p_e - p_m)$ solutions are produced by applying the crossover operator between a solution of the elite set and a solution of the non-elite set, which are both randomly chosen.

FIGURE 1. BRKGA – transition between generations g and $g + 1$.

Notice that this procedure tends to keep some nice characteristics of the elite set to the solutions of the next generation. Figure 1 illustrates two generations in a row of BRKGA with the application of these operators, which is carried out during a fixed number of generations (maxGEN) of the BRKGA. Furthermore, the best solution is obtained at the end of all generations. For detailed information about this method, see [25, 41].

3.3. STRATMH heuristic

Aiming to find reasonable quality solutions for the stratification problem associated with optimization Problem (II), we present a heuristic algorithm based on the BRKGA method and the discretization presented in Section 3.1. The proposed heuristic, called STRATMH, differs from the BRKGA in terms of the forms of representation and construction of the solutions that make up the population evaluated in each generation and the crossover operator.

It is worth mentioning that the way we carry out a population from one generation to the next one, as described in Section 3.2 and represented in Figure 1, is the main difference of the heuristic developed here if compared to the genetic algorithm paradigm, as implemented in Ballin and Barcarolli [3].

3.3.1. Representation and construction of solutions

Each solution built by the heuristic is represented by a vector w with L values w_h , corresponding to the number of observations of B in the h th stratum, such that the constraints (14) and (15), presented in Section 3.1 and here again, are fulfilled:

$$w_1 + \dots + w_h + \dots + w_L = |B|, \quad (19)$$

$$w_h \geq 2, \quad h = 1, \dots, L. \quad (20)$$

In order to ensure that every solution built satisfies these restrictions, the following procedure applies:

- (i) Select a random value $w_1 \in \{2, 3, \dots, |B| - 2(L - 1)\}$.
- (ii) Randomly select the number of observations from the h th stratum; that is, choose a random value $w_h \in \{2, \dots, (|B| - \sum_{i=1}^{h-1} w_i - 2(L - h))\}$ for each $h = 2, \dots, L - 1$.
- (iii) Calculate the value of w_L using $|B| - \sum_{h=1}^{L-1} w_h$.

In this procedure, except for the last stratum, the choice number in the other strata is defined based on a random selection, considering the minimum (same) and maximum values per stratum. It is possible to verify

TABLE 1. Examples of solutions produced.

Selected w_h values	Solution 1 (w)	Solution 2 (w)
$w_1 \in \{2, \dots, 9\}$	5	3
$w_2 \in \{2, \dots, 15 - w_1 - 4\}$	3	6
$w_3 \in \{2, \dots, 15 - w_1 - w_2 - 2\}$	4	3
$w_4 = B - w_1 - w_2 - w_3$	3	3

that the use of (i), (ii), and (iii) guaranteed the fulfillment of (19) and (20). The satisfaction of constraint (19) is given by the definition of step (iii). In the case of the restrictions in (20), by the definition of (i) and by the recurrence relation in (ii), it is observed that each w_h ($h = 1, \dots, L - 1$) generated randomly will be greater than or equal to 2; missing to verify that $w_L \geq 2$ also occurs. Since $w_h \geq 2$ ($h = 1, \dots, L - 1$), we have that $w_1 + w_2 + \dots + w_{L-1} \geq 2x(L - 1)$. Finally, considering this last inequality and the hypothesis that $|B| \geq 2L$ (as per Sect. 3.1), it has to $w_L = B - (w_1 + \dots + w_{L-1}) \geq 2L - 2(L - 1) \geq 2$.

Table 1 presents two feasible solutions concerning (19) and (20) produced from applying the procedure mentioned previously, considering $L = 4$ and $|B| = 15$.

From this procedure, each w vector generated corresponds to a possible X_U stratification. More specifically, considering each vector w , the values of N_h and S_{hx}^2 are calculated, which are used in equation (13), together with the target cv (cv_t), in order to obtain the value of n , the objective function of the optimization model defined in Problem II. This procedure is used in the first generation (iteration) of the heuristic to produce the p initial w vectors (stratifications), and as a mutation operator, to produce, in the other generations, a quantity p_m of new solutions, that is, vectors with possible stratifications.

3.3.2. Crossover and mutation operators

In each generation, after calculating the objective function (sample size) for each of the p vectors w , these vectors are arranged in a non-decreasing order. Then, an elite set denoted by S_e is defined, consisting of p and w vectors corresponding to the p_e smallest sample sizes, with the $(p - p_e)$ remaining vectors associated with the non-elite set denoted by S_{ne} . Finally, following the same principle as BRKGA, the elite set vectors, corresponding to the best stratifications, are added to the next generation's population, together with p_m mutant vectors – corresponding to new stratifications obtained by applying the construction procedure described in Section 3.3.1.

In order to obtain the $(p - p_e - p_m)$ remaining solutions evaluated in the next generation, is applied a crossover operator for each pair (w_e, w_{ne}) , obtained from the random selection of a vector $w_e \in S_e$ and a vector $w_{ne} \in S_{ne}$. As for a crossed pair, two new vectors (stratifications) are produced; this operator must be applied $(p - p_e - p_m)/2$ times.

The crossover exchanges the values of each of the corresponding L positions of the chromosomes w_e and w_{ne} , redistributing the absolute difference d , such that $d = |w_e[i] - w_{ne}[i]|$ ($i = 1, \dots, L$), between these values, proportionally to the values of w_e and w_{ne} associated with the other strata, adding or subtracting such difference by the other positions of this vector. At each exchange and redistribution, two new stratifications are produced for which the objective function's value is calculated. At the end of the L exchanges, the two best solutions produced (with smaller sample sizes), among the $2L$ solutions, are added to the set formed by the $(p - p_e - p_m)$ solutions of the crossover.

In the application of the crossover operator, carried out in L iterations, the values of w_{ne} and w_e , associated with the i th position corresponding to the i th iteration, are exchanged, being necessary to update the values in the other positions of these vectors, so that the sums of the elements of w_e and w_{ne} are equal to $|B|$, guaranteeing the feasibility of the solutions. More specifically, new child chromosomes are produced with each exchange and

update, denoted by w_1 and w_2 . Such exchanges imply a change in the number of observations of the strata, that is, in new stratifications of the population defined from the stratification variable X .

Now, define $s = +1$ if $w_e - w_{ne} > 0$, and $s = -1$, otherwise. In each change, always based on the elements of w_e and w_{ne} that are in the i th position, and to update, we apply the expressions: $\text{round}\left[w_e[k] \times \left(1 + s \frac{d}{(|B| - w_e[i])}\right)\right]$ and $\text{round}\left[w_{ne}[k] \times \left(1 + s \frac{d}{(|B| - w_{ne}[i])}\right)\right]$ for all $k \neq i$, to produce the remaining chromosome elements that correspond to new child vectors.

Considering $L = 3$ and $|B| = 300$, we illustrate the crossover operator applied to $w_e = \{100, 150, 50\}$ and $w_{ne} = \{80, 200, 20\}$. In the first iteration, $w_e[1]$ is exchanged with $w_{ne}[1]$, calculating, in sequence, $|w_e[1] - w_{ne}[1]| = d = 20$. Then, the other positions (2nd and 3rd) of the vectors w_1 and w_2 are obtained as the following:

– for $k = 2$, we obtain

$$w_1[2] = \text{round}\left[150 \times \left(1 + \frac{20}{200}\right)\right] = 165,$$

and

$$w_2[2] = \text{round}\left[200 \times \left(1 - \frac{20}{220}\right)\right] = 182.$$

– for $k = 3$, we obtain

$$w_1[3] = \text{round}\left[50 \times \left(1 + \frac{20}{200}\right)\right] = 55,$$

and

$$w_2[3] = \text{round}\left[20 \times \left(1 - \frac{20}{280}\right)\right] = 18.$$

Thus, after the first iteration, we get $w_1 = \{80, 165, 55\}$ and $w_2 = \{100, 82, 18\}$. In the 2nd iteration, $w_e[2]$ is exchanged by $w_{ne}[2]$, and $|w_e[2] - w_{ne}[2]| = d = 50$. Then, the other positions (1st and 3rd) of w_1 and w_2 are obtained as the following:

– for $k = 1$, we obtain

$$w_1[1] = \text{round}\left[100 \times \left(1 - \frac{50}{150}\right)\right] = 67,$$

and

$$w_2[1] = \text{round}\left[80 \times \left(1 + \frac{50}{100}\right)\right] = 120.$$

– for $k = 3$, we obtain

$$w_1[3] = \text{round}\left[50 \times \left(1 - \frac{50}{150}\right)\right] = 33,$$

and

$$w_2[3] = \text{round}\left[20 \times \left(1 + \frac{50}{100}\right)\right] = 30.$$

Thus, $w_1 = \{67, 200, 33\}$ and $w_2 = \{120, 150, 30\}$ (2nd change). Analogous operations are performed in the 3rd iteration. Figure 2 summarizes the application of the crossover operator to this example. Also, in Appendix A, Algorithm 2 presents the pseudocode that implements all steps of this operator.

		Strata			
Vectors		1	2	3	B
w_e		100	150	50	300
w_{ne}		80	200	20	

1st Change		Strata			
Vectors		1	2	3	d
w_1	80	165	55		B - $w_e[1]$ = 200 B - $w_{ne}[1]$ = 220
w_2	100	182	18		

		Operations	
		$w_1[i] = \text{round}(w_e[i](1+d/200)); i \neq 1$	
		$w_2[i] = \text{round}(w_{ne}[i](1-d/220)); i \neq 1$	

2nd Change		Strata			
Vectors		1	2	3	d
w_1	67	200	33		B - $w_e[2]$ = 150 B - $w_{ne}[2]$ = 100
w_2	120	150	30		

		Operations	
		$w_1[i] = \text{round}(w_e[i](1-d/150)); i \neq 2$	
		$w_2[i] = \text{round}(w_{ne}[i](1+d/100)); i \neq 2$	

3rd Change		Strata			
Vectors		1	2	3	d
w_1	112	168	20		B - $w_e[3]$ = 250 B - $w_{ne}[3]$ = 280
w_2	71	179	50		

		Operations	
		$w_1[i] = \text{round}(w_e[i](1+d/250)); i \neq 3$	
		$w_2[i] = \text{round}(w_{ne}[i](1-d/280)); i \neq 3$	

New Stratifications

80	165	55
100	182	18
67	200	33
120	150	30
112	168	20
71	179	50

W =

FIGURE 2. Crossover operator applied to an example.

Algorithm 1 brings the complete pseudocode of the heuristic. Figure 3 illustrates the working of Algorithm 1 for a random instance, where we considered a population of size $p = 10$, and the same values of L and $|B|$ of the example of Figure 2.

The input parameters are: L (Number of Strata), cv_t (target cv), p (number of vectors w – stratification), PE (percentage of solutions in S_e), PM (percentage of mutant solutions) and maxGEN (maximum number of generations).

About Algorithm 1, in lines 1 and 2, the percentages of elite solutions and solutions built on mutation are defined. In line 3, the initial p solutions constructed are stored in a W_{pxL} matrix. In each generation, between lines 7 and 20, new solutions are produced by evaluating the objective function (calculation of sample size) and applying the crossover and mutation operators. In line 9, the corresponding values of N_h , S_{hx}^2 are calculated for each line of W (w vector). Then, using equations (13) and (12), n and n_h are calculated and rounded. If the values of n_h do not satisfy the constraints (9) and (10) of Problem (II), configuring an infeasibility, the exact method proposed in [9] is applied to obtain integer n and n_h ; otherwise, rounded values are considered.

Algorithm 1. STRATMH (L, cvt, p, PE, PM, maxGEN).

```

1:  $p_e \leftarrow \text{round}(p.\text{PE})$ 
2:  $p_m \leftarrow \text{round}(p.\text{PM})$ 
3: Assign to the matrix  $W$  the  $p$  solutions obtained from the construction procedure
4:  $n_{\text{best}} \leftarrow +\infty$ 
5:  $ng \leftarrow 0$ 
6:  $iterbest \leftarrow 0$ 
7: while ( $ng < \text{maxGEN}$ ) and ( $ng - iterbest \leq \text{round}(0.3 * \text{maxGEN})$ ) do
8:    $ng \leftarrow ng + 1$ 
9:   Calculate, for each solution of  $W$ ,  $N_h$ ,  $S_{hx}^2$  and the value of  $n_h$  and  $n$  (using (12), (13) and exact method)
10:  Sort the solutions (lines) of  $W$  increasingly relative to the  $n$  values
11:  Assign to  $S_e$  the first  $p_e$  lines (best solutions) of  $W$ 
12:  if (sample size ( $n$ ) of the best solution of  $S_e$  is less than  $n_{\text{best}}$ ) then
13:    Assign  $n$  to  $n_{\text{best}}$  and store the associated  $N_h$ ,  $S_{hx}^2$  and  $n_h$  values in  $s_{\text{best}}$ 
14:     $iterbest \leftarrow ng$ 
15:  end if
16:  Assign to  $S_{ne}$  the  $(p - p_e)$  remaining solutions of  $W$ 
17:  Generate, through the construction procedure,  $p_m$  solutions and assign to  $W'$ 
18:  Crossover  $(p - p_e - p_m)/2$  times using  $S_e$  and  $S_{ne}$  solutions (obtaining matrix  $W''$ )
19:   $W \leftarrow S_e \cup W' \cup W''$ 
20: end while
21: Apply the Exact method using  $cvt$ ,  $N_h$  and  $S_{hx}^2$  ( $s_{\text{best}}$ ). Update  $n$  and  $s_{\text{best}}$ .
22: Return  $n$  and  $s_{\text{best}}$ 

```

Generation 1			Generation 2			Generation 2				
Initial Population			Sorted Population			Operators				
100	150	50	S _e	70	40	190	Copy S _e	70	40	190
50	50	200		100	150	50		100	150	50
100	80	120		35	155	110	Mutation (W')	40	40	220
80	200	20	20	80	200	150		100	50	
70	40	190	S _{ne}	150	50	100	Crossover (W'')	80	165	55
40	80	180		50	50	200		100	182	18
100	100	100		100	80	120		133	100	67
150	50	100		80	200	20		75	150	75
20	80	200		40	80	180		50	43	207
35	155	110		100	100	100		70	46	184
Calculate Objective Function										

This method uses as an input data the values of N_h , S_{hx}^2 and cv_t , producing as a result, sample sizes $n_h (h = 1, \dots, L)$ in which simultaneously minimize the objective function of Problem (II) and meet your restrictions. This method is applied in the case of infeasibility and in the best solution produced in the last generation of the heuristic because it significantly increases the computational cost of the heuristic if applied in all solutions.

To define the sets S_e and S_{ne} (lines 11 and 16), the lines of W are ordered about the values of n obtained for each stratification. Between lines 12 and 15, the best update occurs solution to the problem. Then, to produce a new set of solutions evaluated in the population (matrix W) of the next generation, the crossover and mutation operators (lines 17 and 18) are used. Finally, considering the elite S_e solutions (line 11) and the solutions W' (mutation) and W'' (crossover) obtained, respectively, the new population is defined (line 19).

The heuristic is performed until the maximum number of generations is reached (maxGEN) or until there is no reduction in the value of the objective function (sample size) for a pre-defined amount of generations (30% of the total generations as identified in preliminary experiments). Finally, in line 21, the exact method proposed in [9] is applied to the best solution obtained at the end of all generations.

4. NUMERICAL EXPERIMENTS

This section presents the computational results where the STRATMH algorithm and four algorithms from the literature were evaluated: (i) Generalization of the Lavallée and Hidirolou method [45] of strata construction – it can be used with (i) Sethi's (LHSE) [47] or (ii) Kozak's [35] algorithm (LHKO); (iii) VNS algorithm [12] and (iv) genetic algorithm proposed by Balin and Barcarolli (BB) [3]. These four algorithms are implemented and available in R packages, and the STRATMH algorithm, implemented in R, is available in GitHub (StratMH function – available in StratMH package [13]), as shown in Table 2. In addition, Appendix B brings examples of the use of STRATMH.

The remainder of this section provides information about the populations used in the experiments, and Section 4.1 presents the results and analysis of the experiments. In the experiments reported in this section, 31 populations described in Table 3 were used, 28 of them from the literature used in [3, 10–12, 37], and available in the packages of the R language: GA4Stratification (although removed from the CRAN, can be found at: <https://cran.r-project.org/src/contrib/Archive/GA4Stratification/>), stratification [2], stratifyR [43], and sampling (available on the R CRAN). In addition, the authors of this article generated three new populations using functions from the EnvStats package.

Table 3 presents information on populations and Table 4 presents some associated descriptive statistics, such as name, population size (N), the cardinality of B , minimum and maximum values for variable X , and the value of its asymmetry coefficient (AS).

TABLE 2. Information about functions used in algorithms.

Algorithm name	Package or repository	Function name
LHSE (Sethi's algorithm)	stratification	strata.LH
LHKO (Kozak's algorithm)	stratification	strata.LH
VNS	stratvns	STRATVNS
BB	SamplingStrata	optimStrata
STRATMH	https://github.com/jambrito/STRATMH	StratMH

TABLE 3. Description of the 31 populations.

Names	Reference	Description
AgrMinas*	[11]	Agricultural production of municipalities in Minas Gerais State, Brazil (2006)
Anaemia_Haemoglobin	stratifyR	The Anaemia data comes from the Fiji National Nutritional Survey (2004)
BeeFarms	[3, 10–12]	Australian cattle farms stratified by industrial regions
beta103	GA4Stratification	Generated from Beta distribution with parameters $a = 10$ and $b = 3$
CensoCO*	[11]	Data from the 2012 school census in Brazil for the mid-west region
chi1	GA4Stratification	Generated by the Chi-Square distribution with 1 degree of freedom
chi5	GA4Stratification	Generated by the Chi-Square distribution with 5 degree of freedom
Debtors*	Stratification	Debtor population in an Irish firm
GAMMA2000	EnvStats	Generated by the gamma distribution – $\text{rgamma}(2000, 2, 0.5)$
hies.income	stratifyR	Household Income Expenditure Survey (HIES) in Fiji in the year 2010
iso2004	GA4Stratification	Net sales of Turkish industrial companies (2004)
Kozak1, Kozak2, Kozak3*	[10–12, 37]	Generated based on $X = \exp(z)$, where Z is a realization of a normal random variable
ME84	Sampling	Number of municipal employees in 284 municipalities in Sweden (1984)
MRTS*	Stratification	Population simulated from the Monthly Survey on Sales in Retail Trade from Statistics Canada
p100e10	GA4Stratification	Generated by the normal distribution with $\mu = 100$ and $\sigma = 10$
p75	GA4Stratification	Population in thousands of 284 municipalities in Sweden (1975)
pop1076, pop1616, pop2911	[10, 11]	Extracted from the Brazilian Annual Manufacturing Survey (Number of employees)
pop500, pop800	[11, 12]	Generated by the log-normal distribution $X = e^Z$ where Z is normal with $\mu = 4$ and $\sigma^2 = 2.7$
rev84	Sampling	Property values in millions of Swedish kronor from 284 municipalities (1984)
SugarCaneFarms*	[3, 10–12]	Australia's sugar cane farm population
Swiss	Sampling	Information on Swiss municipalities (2003)
TRI900	EnvStats	Generated by the gamma distribution $\text{rttri}(900, \text{min} = 0, \text{max} = 1, \text{mode} = 1/2)$
USbanks	Stratification	Million-dollar funds from major US commercial banks
UScities	Stratification	Population in thousands of American cities (1940)
UScolleges	Stratification	Number of students at four-year US colleges in 1952–1953
WEIBULL1500*	EnvStats	Generated by the weibull distribution – $\text{rweibull}(1500, 0.5)$

Notes. *Populations used in the experiments to calibrate the STRATMH in Section 4.1.

TABLE 4. Basic information of the 31 populations.

Name	N	$ B $	Min	Max	AS
AgrMinas*	844	226	5	47 800	7.30877
anaemia_Haemoglobin	724	90	6.1	16.9	-0.43629
BeefFarms	430	353	50	24250	4.56352
beta103	1000	1000	357.983	985.9574	-0.69695
CensoCO*	9977	79	1	911	40.03439
chi1	1000	1000	4.23E-06	12.98871	2.66591
chi5	1000	1000	0.059874	23.42703	1.40219
Debtors*	3369	1129	40	28 000	6.43743
GAMMA2000	2000	2000	0.056305	16.22303	1.27668
hies_income	3566	3491	34.1224	132921.6	2.88252
iso2004	487	487	63 582 908	1.04E+10	10.02768
Kozak1	4000	51	3	72	1.39607
Kozak2	4000	2837	243	28 578	2.65563
Kozak3*	2000	581	6	2793	3.54417
ME84	284	264	173	47 074	8.63944
MRTS*	2000	2000	141.2	486 366.5	8.60937
p100e10	1000	1000	73.55778	127.3164	-0.03064
p75	284	68	4	671	8.42744
pop1076	1076	88	5	1643	13.23037
pop1616	1616	165	5	2618	11.09524
pop2911	2911	247	5	2497	11.49960
pop500	500	261	1	4 784 142	21.53249
pop800	800	402	1	473 510	22.12974
rev84	284	277	347	59 877	7.83394
SugarCaneFarms*	338	101	18	280	2.26478
Swiss	2896	881	0	3634	2.72729
TRI900	900	900	0.003221	0.97282	-0.01988
USbanks	357	200	70	977	2.06696
UScities	1038	116	10	198	2.86826
UScolleges	677	576	200	9623	2.45168
WEIBULL1500*	1500	1500	3.68E-08	38.25109	4.32759

4.1. Computational results

The experiments with the five algorithms were carried out in RStudio, using a computer with 16 GB of RAM and equipped with an AMD FX-6300 3.5 GHz processor. Regarding the functions presented in Table 2, associated with the four algorithms in the literature, they have used the default values of its parameters.

In the case of the parameters used in the STRATMH heuristic, a preliminary calibration experiment was carried out, based on the analyzes and recommendations made in [25] and taking as a reference a set of values for each of its parameters, as shown in Table 5. Such an experiment is essential since the value considered in the set of parameters is a factor that contributes significantly to the good performance of any algorithm based on a stochastic optimization method.

To determine the ideal combination of the parameters of the proposed heuristic, seven populations were used in this experiment, which we marked with an asterisk in Table 4. The heuristic was performed 10 times to stratify each of these populations, for each $L \in \{3, 4, 5, 6, 7\}$, $cv_t = 5\%$ and $cv_t = 10\%$ and considering 320 combinations of the parameters p , maxGEN, p_e , p_m (Tab. 5) – making, per population, 32 000 executions (number of strata $\times$ target cv $\times$ combinations of parameters $\times$ number of executions), being stored, by execution, the final value of the objective function, that is, smaller sample size.

TABLE 5. STRATMH parameters for calibration experiments.

Parameter	Description	Values
p	Size of population (number of w vectors used by generation)	20, 30, 40, 50, 100
maxGEN	Number of generations	20, 30, 40, 50
p_e	Size of elite population	$0.15p, 0.2p, 0.25p, 0.30p$
p_m	Size of mutant population	$0.15p, 0.2p, 0.25p, 0.30p$

In the first stage of the experiment, considering each population, the number of strata, and target cv, the best solution was the one associated with the smallest sample size (best solution) obtained in the 3200 runs (combination of parameters $\times$ 10 trials). Then, the ratio between the number of runs that each combination produced a sample size equal to the best solution and the total runs of the heuristic using that combination was calculated – in case 10. The closer to 1 such ratio, the more times the combination produced the best solution.

Then, considering such proportions for each of the populations $\times$ number of strata $\times$ target cv's ($7 \times 5 \times 2$), the proportions were decreasingly ordered, taking, by population, number of strata (L) and target cv's (cv_t), the combinations of parameters associated with the first five proportions in such ordering – making a list of 350 combinations. Finally, considering all the combinations, the most frequent combination in this list was chosen as the winning combination, reaching the combination: $p = 50$, maxGEN = 50, $p_e = 0.3 p = 15$, and $p_m = 0.3 p = 15$.

Considering this combination for the STRATMH algorithm and the default parameters of the four algorithms in the literature, the five algorithms were applied in each of the 31 populations (Tab. 4), being evaluated 20 different scenarios corresponding to the number of strata (L) between 3 and 7 and target cv's (cv_t) of, respectively, 3%, 5%, 7.5%, and 10%, making a total of 620 solutions. In addition, their sample sizes and processing times were evaluated for comparison and analysis of the algorithms.

It was observed that the sample size produced, about 31% of the solutions, when evaluating the solutions produced by the BB algorithm, concerning some populations and scenarios, corresponded to a lower number of strata than that previously established; a fact resulting from this algorithm to favor, in the stratification, the smallest possible sample size to the detriment of the number of strata. Additionally, in the case of the LHSE algorithm, considering the solutions produced for some populations and scenarios, non-convergence was observed in about 5% of the solutions. More specifically, the algorithm produced a sample size whose associated coefficient of variation was greater than the target cv (cv_t).

From these observations, in order to enable a correct comparison between the five algorithms, four types of analysis were performed: (i) the first analysis is performed using the 620 solutions produced by the LHKO, VNS, and STRATMH algorithms; (ii) the second, consider the intersection of the set of 587 valid solutions produced by the LHSE algorithm and the 620 solutions produced by the LHKO, VNS and STRATMH algorithms; (iii) the third considers the analysis of the five algorithms, considering the intersection of sets of valid solutions (total of 396) produced by the BB and LHSE algorithm together with the other algorithms, that is, the cases in which the number of strata produced by the BB algorithm corresponds to the number previously fixed and that the LHSE algorithm converged; (iv) considers an experimental analysis of the BB and STRATMH algorithms by applying hypothesis testing to validate the results statistically.

4.1.1. Analysis I – LHKO, VNS and STRATMH algorithms

Table 6 brings the percentage of best solutions for the three algorithms by the number of strata and target cv's, and Table 7 brings the overall percentage of best solutions for each algorithm. Here, the best solution is the one corresponding to the smallest sample size. From these tables, it is observed that the STRATMH algorithm performed better than the LHKO and VNS algorithms, with percentages of better solutions greater than or equal (in bold) to these two algorithms in 19 of the 20 analyzed scenarios, with a percentage lower only

TABLE 6. Percentage of best solutions produced by algorithm, number of strata and cv_t .

Algorithm	$cv_t = 3.0\%$					$cv_t = 5.0\%$				
	L					L				
	3	4	5	6	7	3	4	5	6	7
LHKO	87.1	71.0	67.7	64.5	35.5	83.9	87.1	67.7	74.2	51.6
STRATMH	100.0	93.5	93.5	96.8	80.6	96.8	100.0	96.8	90.3	96.8
VNS	100.0	90.3	90.3	80.6	64.5	96.8	87.1	90.3	90.3	77.4
Algorithm	$cv_t = 7.5\%$					$cv_t = 10.0\%$				
	L					L				
	3	4	5	6	7	3	4	5	6	7
LHKO	87.1	64.5	64.5	74.2	83.9	77.4	74.2	87.1	87.1	96.8
STRATMH	100.0	100.0	93.5	96.8	96.8	100.0	100.0	96.8	96.8	93.5
VNS	100.0	96.8	87.1	93.5	87.1	100.0	96.8	93.5	96.8	100.0

TABLE 7. Percentage of times that algorithm produced the best solution.

Algorithm	% Times best
STRATMH	96.0
VNS	91.0
LHKO	74.4

than that of the algorithm VNS in the scenario associated with $L = 7$ and $cv_t = 10\%$. Considering the overall percentage of solutions (Tab. 7), STRATMH reached 96.0%, followed by the VNS algorithms with 91.0% and LHKO with 74.4%.

In addition to the percentages of best solutions, the relative efficiency (Eff) of the STRATMH algorithm was evaluated about the VNS and LHKO algorithms, that is, the ratio between the sample sizes produced. Considering each of the 620 sample sizes (solutions) obtained, the value of expressions in (21) and (22) was calculated for each population and scenario (L and cv_t), with j being the index corresponding to the j th solution and n_j^{LHKO} , n_j^{VNS} and n_j^{STRATMH} correspond to the minimum sample sizes to reach the predefined target cv , considering L strata and the LHKO, VNS, and STRATMH algorithms. The higher the Eff value, the more significant the difference between the sample sizes produced by the STRATMH algorithm relative to these algorithms.

$$\text{Eff}(\text{LHKO}, \text{STRATMH}) = \frac{n_j^{\text{LHKO}}}{n_j^{\text{STRATMH}}}, \quad j = 1, \dots, 620 \quad (21)$$

$$\text{Eff}(\text{VNS}, \text{STRATMH}) = \frac{n_j^{\text{VNS}}}{n_j^{\text{STRATMH}}}, \quad j = 1, \dots, 620. \quad (22)$$

These values were used to build the boxplots presented in Figures 4 and 5, which show the distributions of the relative efficiencies of the STRATMH algorithm about the LHKO and VNS algorithms by the number of strata and coefficients of target variation. From the analysis of these figures, it is observed that the STRATMH algorithm's most significant gains were concerning the LHKO algorithm. Furthermore, the STRATMH algorithm, in most cases, has a relative efficiency better than or equal to these algorithms. Finally, the smaller the value of cv_t , that is, when a higher level of precision is required, the greater the gains of the STRATMH algorithm.

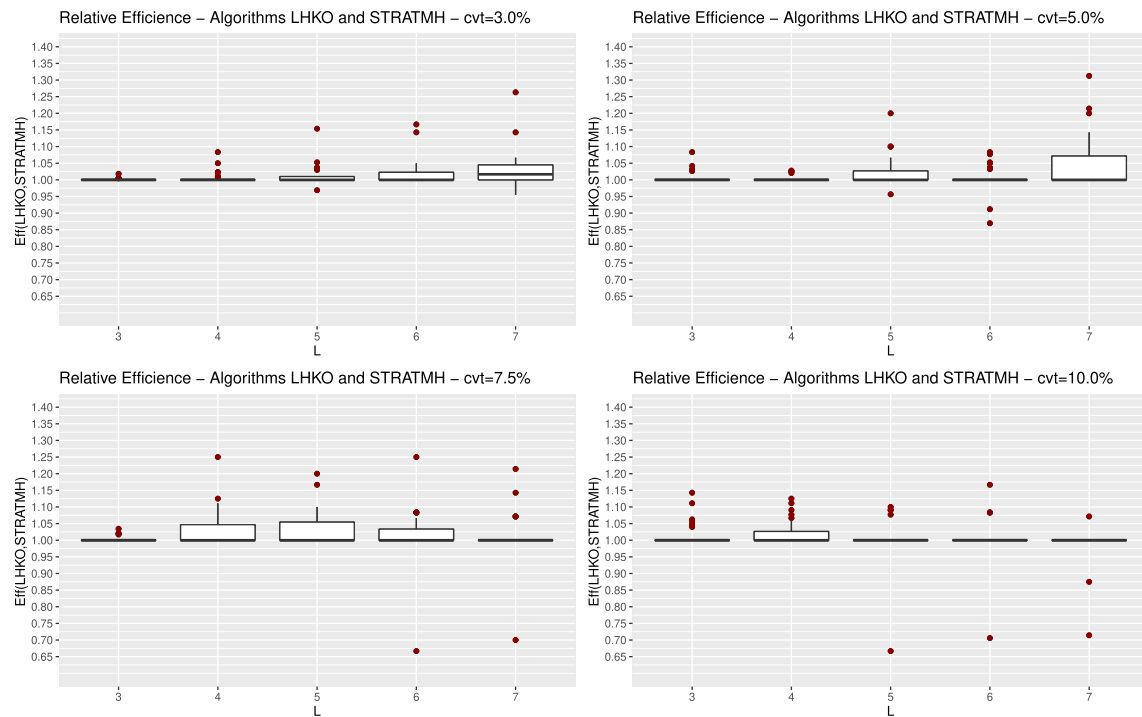


FIGURE 4. Relative Efficiencies by strata and target cv (cv_t) - Algorithms LHKO, STRATMH.

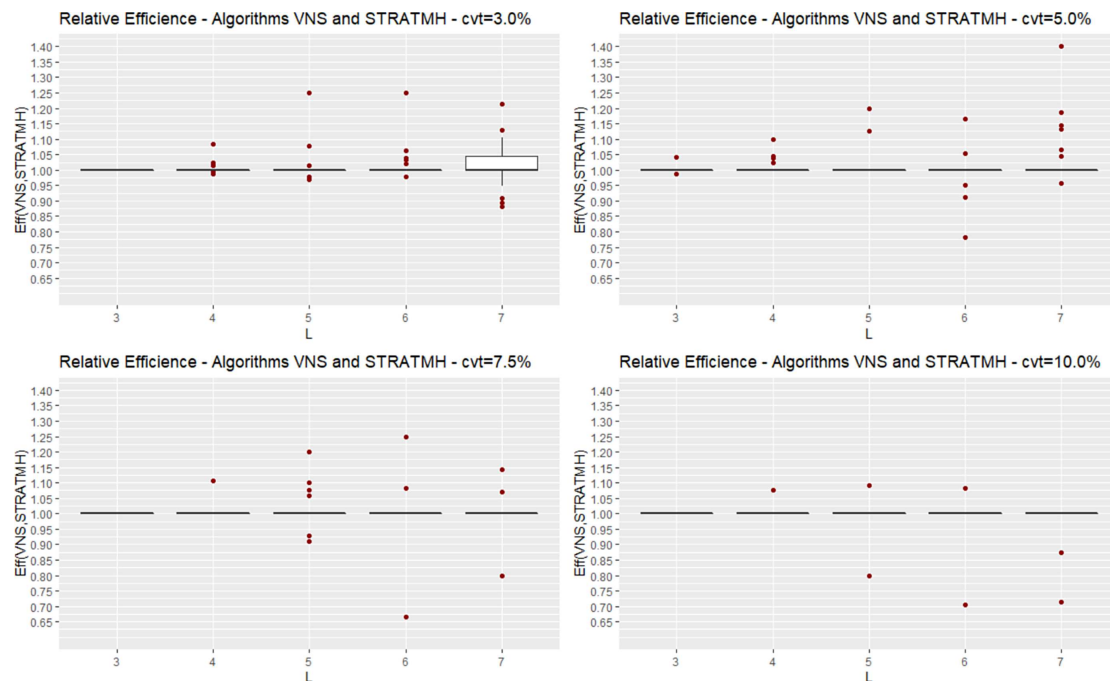


FIGURE 5. Relative Efficiencies by strata and target cv (cv_t) - Algorithms VNS, STRATMH.

TABLE 8. Percentage of best solutions produced by algorithm, number of strata and cv_t .

Algorithm	$cv_t = 3.0\%$					$cv_t = 5.0\%$				
	L					L				
	3	4	5	6	7	3	4	5	6	7
LHSE	33.3	21.4	10.3	15.4	14.3	33.3	21.4	14.3	24.1	37.9
LHKO	86.7	67.9	65.5	69.2	32.1	83.3	85.7	64.3	75.9	51.7
STRATMH	100.0	92.9	93.1	96.2	78.6	96.7	100.0	96.4	89.7	96.6
VNS	100.0	89.3	89.7	84.6	71.4	96.7	85.7	89.3	89.7	79.3
Npops*	30	28	29	26	28	30	28	28	29	29
Algorithm	$cv_t = 7.5\%$					$cv_t = 10.0\%$				
	L					L				
	3	4	5	6	7	3	4	5	6	7
LHSE	29.0	23.3	40.0	43.3	53.3	32.3	36.7	63.3	73.3	76.7
LHKO	87.1	63.3	66.7	76.7	86.7	77.4	76.7	90.0	90.0	96.7
STRATMH	100.0	100.0	93.3	96.7	96.7	100.0	100.0	96.7	96.7	96.7
VNS	100.0	96.7	86.7	96.7	86.7	100.0	96.7	93.3	100.0	100.0
Npops*	31	30	30	30	30	31	30	30	30	30

Notes. *Number of populations where LHSE algorithm converged.

TABLE 9. Percentage of times that algorithm produced the best solution.

Algorithm	% Times best
STRATMH	95.9
VNS	91.8
LHKO	75.0
LHSE	35.3

4.1.2. Analysis II – LHKO, LHSE, VNS and STRATMH algorithms

Table 8 brings, by the number of strata and target cv 's, the percentage of best solutions concerning the sample sizes produced by the LHKO, LHSE, VNS, and STRATMH algorithms, and Table 9 brings the overall percentage of each of the algorithms, considering all the 587 valid solutions produced by the LHSE algorithm. From Table 8, it can be seen that, in 18 of the 20 scenarios (in bold), STRATMH produced percentages greater than or equal to those of the other algorithms, followed by the VNS and LHKO algorithms. The scenarios where STRATMH performed less than VNS correspond to $L = 6, 7$, and $cv_t = 10\%$. Additionally, the algorithm produced, in 18 of the 20 scenarios, a percentage of the best solutions above 93%, with only a percentage of the order of 79% ($L = 7$ and $cv_t = 3\%$) and a percentage of the order of 90 being observed % ($L = 6$ and $cv_t = 5\%$). Furthermore, compared to the second-best algorithm (VNS), STRATMH produced the highest percentages of best solutions in 12 of the 20 scenarios. Finally, the biggest gains of this algorithm about the other algorithms, translated in percentage terms, are observed for $cv_t = 7.5\%$.

In Table 9, note that STRATMH has superior gain with an overall percentage of the order of 96.0% of the best solutions, followed by the VNS algorithms with around 92.0% and LHKO with 75.0%.

4.1.3. Analysis III – LHKO, LHSE, VNS, BB and STRATMH algorithms

In this last analysis, five algorithms were evaluated considering only the feasible solutions produced by the BB and LHSE algorithms (a total of 396). From the analysis of Table 10, which brings the percentage of best solutions per algorithm, the number of strata, and cv_t , and Table 11, which brings the overall percentage of best solutions for each of the algorithms, it is observed that the STRATMH algorithm presented, again, superior

TABLE 10. Percentage of best solutions produced by algorithm, number of strata and cv_t .

Algorithm	$cv_t = 3.0\%$					$cv_t = 5.0\%$				
	L					L				
	3	4	5	6	7	3	4	5	6	7
BB	44.8	32.0	15.4	13.6	17.4	63.0	60.0	40.9	22.2	35.3
LHKO	82.8	64.0	61.5	68.2	21.7	77.8	84.0	59.1	61.1	29.4
LHSE	27.6	12.0	0.0	4.5	0.0	25.9	12.0	0.0	11.1	17.6
STRATMH	96.6	92.0	92.3	95.5	73.9	92.6	100.0	95.5	83.3	94.1
VNS	96.6	88.0	88.5	86.4	69.6	92.6	84.0	90.9	88.9	70.6
Npops*	29	25	26	22	23	27	25	22	18	17
Algorithm	$cv_t = 7.5\%$					$cv_t = 10.0\%$				
	L					L				
	3	4	5	6	7	3	4	5	6	7
BB	78.6	80.8	42.9	46.2	80.0	92.6	81.8	85.7	66.7	–
LHKO	82.1	57.7	52.4	46.2	40.0	74.1	63.6	78.6	83.3	–
LHSE	17.9	11.5	23.8	7.7	0.0	22.2	18.2	50.0	66.7	–
STRATMH	96.4	100.0	90.5	100.0	100.0	100.0	95.5	100.0	100.0	–
VNS	96.4	96.2	85.7	92.3	60.0	100.0	90.9	92.9	100.0	–
Npops*	28	26	21	13	5	27	22	14	6	0

Notes. *Number of populations where BB algorithm produced a feasible solution, that is, attending constraints of the fixed strata number, and the LHSE algorithm converged; – BB algorithm did not produce feasible solution.

TABLE 11. Percentage of times that algorithm produced the best solution.

Algorithm	% Times best
STRATMH	94.2
VNS	89.1
LHKO	64.1
BB	51.5
LHSE	15.7

performance to other algorithms. Table 10 shows, in 18 of the scenarios (in bold), percentages of best solutions (greater than or equal to the other algorithms), with STRATMH showing a lower percentage only in the scenario associated with $L = 6$ and $cv_t = 5\%$, when compared to VNS, which had the second-best performance. Regarding the overall percentage of best solutions Table 11, the STRATMH algorithm had again a percentage above 94%, followed by the VNS (89.1%), LHKO (64.1%), BB (51.5%), and LHSE (15.7%) algorithms.

4.1.4. Analysis IV – Experimental Analysis – BB and STRATMH algorithms

An additional experiment was carried out to evaluate the STRATMH and BB (Ballin and Barcarolli) algorithms by applying hypothesis testing to validate the results statistically. More specifically, the Student's t -test [48] was used to verify whether there is a significant difference between the STRATMH and BB algorithms regarding the quality of the solutions produced. In the t -test, the null hypothesis (H_0) states that the means of two populations are the same, while the alternative (H_1) states that the means are different. Considering the mean and standard deviation of the results of the function to the objective of the stratification problem, that is, the sample sizes produced, the t -test (bilateral) can be applied to verify if the difference between the STRATMH and BB algorithms is significant for a particular population and the values of L and cv_t considered in the experiments and a significance level of 5%. If the difference is significant and STRATMH presents a

TABLE 12. Percentage of Better (B), Equivalent (E) and Worse (W) solutions of the STRATMH algorithm – General.

	B	E	W
General ($P = 408$)	71.1	23.5	5.4

Notes. P = Number of feasible solutions produced by BB algorithm.

TABLE 13. Percentage of Better (B), Equivalent (E) and Worse (W) solutions of the STRATMH algorithm – 20 Scenarios.

Strata and cv_t	P	B	E	W	Strata and cv_t	P	B	E	W
$L = 3, cv_t = 3\%$	30	73.3	20.0	6.7	$L = 3, cv_t = 7.5\%$	28	32.1	53.6	14.3
$L = 4, cv_t = 3\%$	28	92.9	7.1	0.0	$L = 4, cv_t = 7.5\%$	26	57.7	34.6	7.7
$L = 5, cv_t = 3\%$	28	92.9	7.1	0.0	$L = 5, cv_t = 7.5\%$	20	75.0	25.0	0.0
$L = 6, cv_t = 3\%$	27	96.3	3.7	0.0	$L = 6, cv_t = 7.5\%$	12	83.3	16.7	0.0
$L = 7, cv_t = 3\%$	23	100.0	0.0	0.0	$L = 7, cv_t = 7.5\%$	3	100.0	0.0	0.0
$L = 3, cv_t = 5\%$	28	60.7	32.1	7.1	$L = 3, cv_t = 10\%$	28	14.3	67.9	17.9
$L = 4, cv_t = 5\%$	28	82.1	14.3	3.6	$L = 4, cv_t = 10\%$	24	37.5	50.0	12.5
$L = 5, cv_t = 5\%$	26	88.5	11.5	0.0	$L = 5, cv_t = 10\%$	13	46.2	38.5	15.4
$L = 6, cv_t = 5\%$	21	95.2	4.8	0.0	$L = 6, cv_t = 10\%$	2	50.0	0.0	50.0
$L = 7, cv_t = 5\%$	13	92.3	7.7	0.0	$L = 7, cv_t = 10\%$	0	–	–	–

Notes. P = Number of feasible solutions produced by BB algorithm; – BB algorithm did not produce feasible solution.

better (or worse) result (sample size), then it is said to be significantly better (or worse) for a given population about the BB algorithm; otherwise (p -value $> 5\%$), it is said that the algorithms present equivalent (or similar) performances.

The STRATMH and BB algorithms were executed 30 times, based on each of the 31 populations and the 20 scenarios of the previous experiments (combination of $L = 3, 4, 5, 6, 7$ and $cv_t = 3\%, 5\%, 7.5\%$, and 10%) – making a total of 37 200 ($2 \times 30 \times 31 \times 20$) executions. For each population and each scenario, the values of n obtained in the 30 executions of the two algorithms were stored.

From the application of this test, where, initially, we would have, per scenario, 31 evaluated populations, making a total of 620 solutions (in each solution 30 executions), discarding the cases (population $\times$ combination of L and cv_t) where the BB algorithm did not converge (did not produce feasible results), 408 solutions were evaluated. Tables 12 and 13 bring the percentages of the best, equivalent, and worst solutions of the STRATMH algorithm in aggregate form (general) and by scenario. From these tables, it is possible to observe a significant superiority of the STRATMH over the BB algorithm regarding the percentage of significantly better solutions. In particular, the highest percentages of better solutions are observed for target cvs of 3% and 5%. Furthermore, among 20 evaluated scenarios, only in one case ($L = 3$ and $cv_t = 10\%$) the BB algorithm performed better than the STRATMH algorithm.

Finally, Table 14 presents the average and median processing times of the STRATMH, BB, and VNS algorithms. Such analysis was restricted to the three algorithms. Although they are based on stochastic optimization methods (metaheuristics), they demand intensive computation, implying more processing time to produce reasonable quality solutions. The LHSE and LHKO algorithms are faster, with execution time of the order of a few seconds; however, in general, they produced solutions of lower quality than those achieved by STRATMH and VNS, as observed in Tables 6–11.

From the analysis of Table 14, it can be observed that STRATMH is the algorithm that generally requires the least computational time to produce the smallest sample size. The median STRATMH time was less than 4 s, regardless of the scenario. Likewise, a value of fewer than 5 s in average time was observed, at less than $cv_t = 3\%$.

APPENDIX A. CROSSOVER PROCEDURE

Algorithm 2 brings pseudocode with all the steps of the crossover procedure described in Section 3.3.2. In steps 1 and 2, the matrices where the vectors w corresponding to the new stratifications, produced between lines 3 and 28, are defined. Then, in the main loop from line 3 are selected (lines 4 and 5) vectors from the sets S_e and S_{ne} , and these vectors (in lines 6 and 7) are copied to vectors w_1 and w_2 , which will be changed/updated for each change of values of each of their L positions, from line 8 (intermediate loop). Next, between lines 12 and 19, the values of each of the positions of w_1 and w_2 are swapped, and the $(L - 1)$ remaining values are updated, as per lines 13–16 (inner loop). Finally, in line 20, the new vectors produced are added to matrix M and then reset with the initial values of vectors w_e and w_{ne} . At the end of the intermediate loop, after line 24, the values of N_h , S_{hx}^2 , and the sample sizes corresponding to the $2 * L$ stratifications associated with M are calculated.

Finally, in line 26, the two best stratifications, with the smallest associated sample sizes, are added to the matrix Q , which contains all the $(p - p_e - p_m)$ solutions obtained from the crossover.

Algorithm 2. Crossover().

```

1: Set  $M_{2L \times L}$ 
2: Set  $Q_{(p-p_e-p_m) \times L}$ 
3: for  $j \leftarrow 1$  to  $((p - p_e - p_m)/2)$  do
4:   Select a vector  $w_e$  from  $S_e$ 
5:   Select a vector  $w_{ne}$  from  $S_{ne}$ 
6:    $w_1 \leftarrow w_e$ 
7:    $w_2 \leftarrow w_{ne}$ 
8:   for  $h \in \{1, \dots, L\}$  do
9:      $d \leftarrow |w_1[h] - w_2[h]|$ 
10:    if  $(d \neq 0)$  then
11:       $s \leftarrow (w_e[h] - w_{ne}[h])/d$ 
12:      for  $i \in \{1, \dots, L\} \setminus h$  do
13:         $df1 \leftarrow \text{round}(d.w_1[i]/(|B| - w_e[h]))$ 
14:         $df2 \leftarrow \text{round}(d.w_2[i]/(|B| - w_{ne}[h]))$ 
15:         $w_1[i] \leftarrow w_1[i] + s.df1$ 
16:         $w_2[i] \leftarrow w_2[i] - s.df2$ 
17:      end for
18:       $w_1[h] \leftarrow w_{ne}[h]$ 
19:       $w_2[h] \leftarrow w_e[h]$ 
20:      Add  $w_1$  and  $w_2$  to  $M$  (vectors with new stratifications)
21:       $w_1 \leftarrow w_e$ 
22:       $w_2 \leftarrow w_{ne}$ 
23:    end if
24:  end for
25:  Calculate  $N_h, S_{hx}^2, Fobj(n)$  for each line of  $M$  and sort the lines of  $M$  – by  $Fobj$ 
26:  Add the first two rows of  $M$  to the  $W''$ 
27:  Initialize  $M$ 
28: end for
29: Return  $W''$ 

```

APPENDIX B. STRATMH FUNCTION

The STRATMH algorithm was implemented in R language and is available in the stratMH function (see Tab. 2), which makes use of the parallel and MultAlloc packages (with the implementation of the allocation method proposed in [9]). Tables B.1 and B.2 bring, respectively, information about the parameters used as input to the function and the results produced from its application. Finally, Table B.3 brings two examples of the application of this function.

TABLE B.1. stratMH function – Input parameters.

Parameters	Description
X	Vector with N observations of the stratification variable
L	Number of strata (default value = 3)
cvt	Target cv (default value = 0.1 – 10%)
p	Number of solutions (w vectors) per generation (default value = 50)
pe	Percentage of chromosomes in Se (default value = 0.3)
pm	Percentage of chromosomes produced in the mutation (default value = 0.3)
maxgen	Maximum Heuristic Generations (default value = 50)
npar	Performs the number of executions of the algorithm equal to npar (default = 1), npar > 1 parallelism

TABLE B.2. Results produced by the stratMH function.

Values	Description
xbest	Vector with L values of w that satisfy equations (14) and (15)
bk	Vector with $L - 1$ cutoff $b_1, b_2, \dots, b_{L-1}$
Nh	Vector with number of observations of X in each stratum L stratum
nh	Vector with sample size allocated to each stratum
Sh2	Vector with population variance concerning to variable X in each of L strata
fobj	Total sample size (n)
cv	Coefficient of variation obtained from the stratification produced
ta	Runtime in seconds
ibest	The generation where was produced the best solution (smallest sample size)

TABLE B.3. Examples of the application of the stratMH function.

Example 1: Population – Debtors
library(StratMH)
s=StratMH(Debtors,L=4,cvt=0.05) #target cv of 5.0%
s
\$xbest
[1] 337 459 261 72
\$bk
[1] 437 1705 6029
\$Nh
[1] 2079 910 308 72
\$nh
[1] 13 16 20 20
\$Sh2
[1] 12219.96 104620.51 1346923.45 24705371.94
\$fobj
[1] 69
\$cv
[1] 0.04999865
\$ta
elapsed
2.8
\$ibest
[1] 33
Example 2: Population – BeefFarms
X=BeefFarms
s=StratMH(X,L=3,cvt=0.01,npar=3) #target cv of 1.0%
s
\$xbest
[1] 160 98 95
\$bk
[1] 310 1016
\$Nh
[1] 225 108 97
\$nh
[1] 17 23 97
\$Sh2
[1] 4894.79939241.212 23587198.247
\$fobj
[1] 137
\$cv
[1] 0.00995282
\$ta
elapsed
14.46
\$ibest
[1] 4

APPENDIX C. STRATIFIED SAMPLING NOTATION

TABLE C.1. Stratified sampling notation.

U	Set of units $\{1, 2, \dots, N\}$ of the population P
U_h	Population strata containing population units in the $U_h \subset U$ and $U_h \neq 0, h = 1, \dots, L$
N	Total population units
N_h	Total population units in the stratum h ($h = 1, \dots, L$)
n	Sample size
n_h	Number of sample units allocated to the stratum h ($h = 1, \dots, L$)
x_i	X value associated with the i -th population unit U ($i \in U$)
s_h	Sample selected in the stratum h , $s_h \subset U_h$ ($h = 1, \dots, L$)
$\bar{x}_h = \sum_{i \in s_h} \frac{x_i}{n_h}$	Mean of the x_i values sampled in the stratum h
$\bar{X}_h = \sum_{i \in U_h} \frac{x_i}{N_h}$	Mean of the x_i values in the stratum h
$T_x = \sum_{h=1}^L \sum_{i \in U_h} x_i$	Total Population
$S_{xh}^2 = \sum_{i \in U_h} \frac{(x_i - \bar{X}_h)^2}{N_h - 1}$	Population Variance of X in the stratum h
S_{xh}	Population Standard deviation of X in the stratum h
$t_x = \sum_{h=1}^L N_h \bar{x}_h$	Total estimator for X
$V(t_x) = \sum_{h=1}^L N_h^2 \frac{S_{xh}^2}{n_h} \left(1 - \frac{n_h}{N_h}\right)$	Total estimator variance
$cv(t_x) = \frac{\sqrt{V(t_x)}}{t_x}$	Coefficient of variation of the total estimator (level of precision)

Acknowledgements. The authors would like to thank the anonymous referee for his/her valuable comments/suggestions that helped us to improve the paper. The research of Leonardo de Lima was partly funded by the CNPq grant number 315739/2021-5. The research of José Brito was partly funded by the CNPq grant number 405044/2021-6.

Data Availability Statement. The research data/code associated with this article are available in <https://github.com/jambrito/STRATMH> [13].

REFERENCES

- [1] S. Baillargeon and L. Rivest, A general algorithm for univariate stratification. *Int. Stat. Rev.* **77** (2009) 331–344.
- [2] S. Baillargeon and L. Rivest, The construction of stratified designs in R with the package stratification. *Surv. Methodol.* **37** (2011) 53–65.
- [3] M. Ballin and G. Barcarolli, Joint determination of optimal stratification and sample allocation using genetic algorithm. *Surv. Methodol.* **39** (2013) 369–393.
- [4] M. Bankier, Power allocation: determining sample sizes for subnational areas. *Am. Stat.* **42** (1988) 174–177.
- [5] C. Blum, J. Puchinger, G.R. Raidl and A. Roli, Hybrid metaheuristics in combinatorial optimization: a survey. *Appl. Soft Comput.* **11** (2011) 4135–4151.
- [6] R. Bouabda, B. Jarboui, M. Eddaly and A. Rebaï, A branch and bound enhanced genetic algorithm for scheduling a flowline manufacturing cell with sequence dependent family setup times. *Comput. Oper. Res.* **38** (2011) 387–393.
- [7] J. Brito, N. Maculan, M. Lila and F. Montenegro, An iterative local search approach applied to the optimal stratification problem. *Int. Trans. Oper. Res.* **17** (2010) 753–764.
- [8] J. Brito, N. Maculan, M. Lila and F. Montenegro, An exact algorithm for the stratification problem with proportional allocation. *Optim. Lett.* **4** (2015) 185–195.

- [9] J. Brito, P. Silva, G. Semaan, N. Maculan, Integer programming formulations applied to optimal allocation in stratified sampling. *Surv. Methodol.* **41** (2015) 427–442.
- [10] J. Brito, G. Semaan, A. Fadel and L. Brito, An optimization approach applied to the optimal stratification problem. *Commun. Stat. Simul. Comput.* **46** (2017) 4419–4451.
- [11] J. Brito, T. Veiga and P. Silva, An optimisation algorithm applied to the one-dimensional stratification problem. *Surv. Methodol.* **45** (2019) 295–315.
- [12] J. Brito, L. de Lima, P. Gonzalez, B. Oliveira and N. Maculan, Heuristic approach applied to the optimum stratification problem. *RAIRO: Oper. Res.* **55** (2021) 979–996.
- [13] J.A. Brito, G. Semaan, L. de Lima and A. Fadel, Algorithm for univariate stratification problem. <https://github.com/jambrito/STRATMH> (2023).
- [14] W. Cochran, Sampling Techniques, 3rd edition. New York, John Wiley (1977).
- [15] E. Creaco and G. Pezzinga, Embedding linear programming in multi objective genetic algorithms for reducing the size of the search space with application to leakage minimization in water distribution networks. *Environ. Modell. Softw.* **69** (2015) 308–318.
- [16] T. Dalenius and J.J. Hodges, Minimum variance stratification. *J. Am. Stat. Assoc.* **54** (1959) 88–101.
- [17] F. Danish, A mathematical programming approach for obtaining optimum strata boundaries using two auxiliary variables under proportional allocation. *Stat. Transition New Ser.* **19** (2018) 507–526.
- [18] F. Danish and S. Rizvi, Optimum stratification in Bivariate auxiliary variables under Neyman allocation. *J. Mod. Appl. Stat. Methods* **17** (2018) 2–24.
- [19] F. Danish, S. Rivsi, M. Sharma and M. Jeelani, Optimum stratification using mathematical programming approach: a review. *J. Stat. App. Probab. Lett.* **4** (2017) 123–129.
- [20] F. Danish, S. Rizvi, M. Sharma, M.I. Jeelani, B. Kumar and Q. Farooq Da, Optimum stratification for two stratifying variables. *Revista Investigacion Operacional* **40** (2019) 562–573.
- [21] F.M. Defersha and S. Bayat Movahed, Linear programming assisted (not embedded) genetic algorithm for flexible jobshop scheduling with lot streaming. *Comput. Ind. Eng.* **117** (2018) 319–335.
- [22] G. Ekman, An approximation useful inf univariate stratification. *Ann. Math. Stat.* **30** (1959) 219–229.
- [23] A. Fadel, L. Ochi, J.A.M. Brito and G. Semaan, Microaggregation heuristic applied to statistical disclosure control. *Inf. Sci.* **548** (2021) 37–55.
- [24] P. Festa, A biased random-key genetic algorithm for data clustering. *BIOCOMP Math. Biosci.* **245** (2013) 76–85.
- [25] J.F. Gonçalves and M.G. Resende, Biased random-key genetic algorithms for combinatorial optimization. *J. Heuristics* **17** (2013) 487–525.
- [26] J.F. Gonçalves and M.G. Resende, A biased random-key genetic algorithm for the unequal area facility layout problem. *Eur. J. Oper. Res.* **246** (2015) 86–107.
- [27] P. Gunning and J. Horgan, A new algorithm for the construction of stratum boundaries in skewed populations. *Surv. Methodol.* **30** (2004) 159–166.
- [28] D. Hedlin, A procedure for stratification by an extended Ekman rule. *J. Official Stat.* **16** (2000) 15–29.
- [29] M. Hidiroglou and M. Kozak, Stratification of skewed populations: a comparison of optimisation-based versus approximate methods. *Int. Stat. Rev.* **86** (2018) 87–105.
- [30] T. Kesintürk and S. Er, A genetic algorithm approach to determine stratum boundaries and sample sizes of each stratum in stratified sampling. *Comput. Stat. Data Anal.* **52** (2007) 53–67.
- [31] E. Khan, M. Khan and M. Ahsan, Optimum stratification: a mathematical programming approach. *Calcutta Stat. Assoc. Bull.* **52** (2002) 323–334.
- [32] M. Khan, N. Nand and N. Ahmad, Determining the optimum strata boundary points using dynamic programming. *Surv. Methodol.* **34** (2008) 205–214.
- [33] M. Khan, D. Rao, A. Ansari and M. Ahsan, Determining optimum strata boundaries and sample sizes for skewed population using log-normal distribution. *Commun. Stat. Simul. Comput.* **44** (2015) 1364–1387.
- [34] M. Khan, K. Reddy and D. Rao, Designing stratified sampling in economic and business surveys. *J. Appl. Stat.* **42** (2015) 2080–2099.
- [35] M. Kozak, Optimal stratification using random search method in agricultural surveys. *Stat. Transition* **6** (2004) 797–806.
- [36] M. Kozak, Comparison of random search method and genetic algorithm for stratification. *Commun. Stat. Simul. Comput.* **43** (2014) 249–253.
- [37] M. Kozak and M. Verma, Geometric versus optimization approach to stratification: a comparison of efficiency. *Surv. Methodol.* **32** (2006) 157–163.
- [38] P. Lavallée and M. Hidiroglou, On the stratification of skewed populations. *Surv. Methodol.* **14** (1988) 33–43.
- [39] J. Lisic, H. Sang, Z. Zhu and S. Zimmer, Optimal stratification and allocation for the june agricultural survey. *J. Official Stat.* **34** (2018) 121–148.
- [40] S. Lohr, Sampling: Design and Analysis, 2nd edition. Chapman & Hall (2019).
- [41] R. Mart, P. Pardalos and M. Resende, Handbook of Heuristics, 1st edition. Springer (2018).
- [42] K. Reddy and M. Khan, Optimal stratification in stratified designs using weibull-distributed auxiliary information. *Commun. Stat. Theory Methods* **48** (2019) 3136–3152.
- [43] K. Reddy and M. Khan, stratifyR: an R package for optimal stratification and sample allocation for univariate populations. *Aust. New Zealand J. Stat.* **62** (2020) 383–405.

- [44] M. Reisi-Nafchi and G. Moslehi, A hybrid genetic and linear programming algorithm for two-agent order acceptance and scheduling problem. *Appl. Soft Comput.* **33** (2015) 37–47.
- [45] L. Rivest, A generalization of the Lavallée and Hidioglou algorithm for stratification in business surveys. *Surv. Methodol.* **28** (2020) 191–198.
- [46] G. Semaan, J.A.M. Brito, I. Coelho, E.F. Silva, A. Fadel, L.S.O. Ochi and N. Maculan, A brief history of heuristics: from bounded rationality to intractability. *IEEE Latin Am. Trans.* **18** (2020) 1975–1986.
- [47] V. Sethi, A note on the optimum stratification of populations for estimating the population means. *Aust. New Zealand J. Stat.* **5** (1963) 20–33.
- [48] M. Triola, Elementary Statistics, 14th edition. Pearson (2021).
- [49] V. Unnithan, The minimum variance boundary points of stratification. *Sankhya* **40** (1978) 60–72.
- [50] Y. Xiao and A. Konak, A genetic algorithm with exact dynamic programming for the green vehicle routing & scheduling problem. *J. Cleaner Prod.* **167** (2017) 1450–1463.



Please help to maintain this journal in open access!

This journal is currently published in open access under the Subscribe to Open model (S2O). We are thankful to our subscribers and supporters for making it possible to publish this journal in open access in the current year, free of charge for authors and readers.

Check with your library that it subscribes to the journal, or consider making a personal donation to the S2O programme by contacting subscribers@edpsciences.org.

More information, including a list of supporters and financial transparency reports, is available at <https://edpsciences.org/en/subscribe-to-open-s2o>.