STATISTICS IN TRANSITION new series, March 2026
Vol. XX, No. X, pp. xx–XX, https://doi.org/10.59139/stattrans-2026-xxx
Received – 15.04.2025; accepted – xx.xx.xxxx

Optimal sample allocation in multivariate stratified sampling: a comparison of deterministic and stochastic optimization algorithms

Dalius Pumputis[1]

Abstract

This study addresses the problem of optimal sample allocation in multivariate stratified sampling, where survey accuracy and cost-efficiency are the key concerns. Two optimization formulations are examined: one aims to minimize the total survey cost subject to constraints on the precision of the estimators of the population totals, while the other seeks to minimize a weighted sum of the relative variances of these estimators, given a fixed total survey budget. Classical and modern optimization approaches are reviewed and evaluated, including Integer Programming Algorithms (IPA), Bethel's Algorithm (BA), Constrained Optimization by Linear Approximations (COBYLA), and three stochastics, namely Generalized Simulated Annealing Algorithm (GSAA), Particle Swarm Optimization (PSOA) and Biased Random-Key Genetic Algorithm (BRKGA). Using synthetic and real-world populations, numerical experiments demonstrate that IPA consistently achieves the global minimum and serves as the benchmark. While BA underperforms, BRKGA emerges as a competitive alternative, closely matching IPA in most scenarios. Results also highlight the impact of variable skewness on allocation efficiency, with real-world datasets being more complex and thus having higher sampling demands. The findings underscore the importance of adaptive, integer-feasible optimization methods for accurate and cost-effective survey design.

Key words: constrained optimization by linear approximations, integer programming, multivariate stratified sampling, optimal sample allocation, stochastic optimization.

1. Introduction

To obtain accurate estimates, surveys often employ stratification of a finite population. This statistical technique involves dividing the survey population into several distinct, non-overlapping, and internally homogeneous groups known as strata. Independent samples are then drawn from each of these groups. When stratified sampling is selected as the sampling method, the initial task is to define the boundaries of the strata.

After the strata boundaries have been established and the total sample size n has been decided, the next step involves allocating the sample size across the strata. Various allocation strategies are available, such as equal, proportional, or Neyman allocation (Neyman 1934). Equal and proportional methods are generally efficient when within-stratum variances are similar. In contrast, the Neyman method is more appropriate when strata vary significantly, as it prioritizes drawing fewer samples from more homogeneous strata and more from those with greater internal variability.

[1] Vilnius Gediminas Technical University (VILNIUS TECH), Lithuania.
E-mail: dalius.pumputis@vilniustech.lt. ORCID: https://orcid.org/0000-0003-0954-0663.

Neyman allocation relies on a formula designed to minimize both the survey cost C and the variance of the estimator for a single study variable. However, modern surveys often focus on multiple variables. In such cases, an allocation optimized for one variable may not be optimal for others, resulting in what is known as the multivariate optimal sample allocation problem.

This issue has been addressed by several researchers, beginning with Yates (1960), who proposed minimizing a weighted sum of the variances of the estimates for all survey variables, under the constraint of a fixed total sample size. Later, Chatterjee (1967) extended this line of inquiry by deriving an expression for the increase in variance when a non-optimal allocation is used, offering a framework for quantifying the deviation from optimality in multivariate settings.

Ahsan and Khan (1982) formulated the multivariate allocation problem with stratum-level overhead costs as a nonlinear program, minimizing total cost subject to variance constraints. Bethel (1985) proposed a convex programming algorithm that is simple to implement and converges efficiently. He later extended this work (Bethel 1989) by incorporating linear variance constraints and deriving optimal allocations using Lagrangian multipliers, providing a practical algorithm with demonstrated convergence.

Subsequent work has focused on obtaining integer and compromise allocations. Khan et al. (1998), Khan and Ahsan (2003), and Khan et al. (2010) developed dynamic and goal programming methods to derive integer-valued, compromise solutions, incorporating auxiliary information where available. Swain (2013) and Varshney et al. (2014) also applied goal programming to balance efficiency and practicality in multivariate settings.

Kadane (2005) introduced a dynamic sampling plan that minimizes variance at every stage, extending Neyman's approach to sequential designs. Brito et al. (2015) proposed a binary integer programming model that offers improved performance over existing algorithms in complex survey scenarios.

Since study variable parameters are often unknown in advance, Dayal (1985) suggested using auxiliary variables correlated with the variable of interest to guide sample allocation, showing that proportional allocation based on such variables can outperform approximations of Neyman allocation. Reddy et al. (2018) used auxiliary data with dynamic programming to optimize stratum boundaries and sample sizes in health surveys, greatly improving estimation efficiency over traditional methods.

While many allocation methods rely on approximations or rounding to achieve practical sample sizes, these approaches may lead to suboptimal or even infeasible results. Wright (2017) addressed these limitations by proposing exact optimal allocation algorithms that avoid common issues with Neyman allocation, such as non-integer solutions, post-rounding inefficiencies, and allocations exceeding stratum sizes. Expanding on this, Wright (2020) developed an exact algorithm with cost and sample size bounds, using cost-weighted function decomposition to offer a flexible, efficient framework that includes several traditional methods as special cases.

Recent studies propose methodological advances for optimum and compromise allocation in multivariate stratified sampling, tackling issues like non-response, cost, uncertainty, fuzzy environments, and practical constraints.

Haq et al. (2020) tackled compromise allocation in multivariate stratified sampling under non-response and fixed costs by converting an integer non-linear problem into a binary goal programming model, solved with flexible fuzzy goals for population mean estimation. Mahfouz et al. (2023) proposed a stochastic compromise allocation model using multi-objective programming to minimize survey cost and stratum variances. Through chance-constrained programming and simulations, they showed it provides the most efficient allocations. Raghav et al. (2023) addressed compromise allocation under response and non-response using multi-objective intuitionistic fuzzy programming with optimistic and pessimistic strategies, demonstrating applicability through simulations in wildlife, agriculture, and marketing surveys. Jalil et al. (2023) proposed a hierarchical multi-level programming model for compromise allocation under non-response and budget constraints, using fuzzy methods to optimize allocations and improve survey efficiency, flexibility, and cost-effectiveness. Gupta et al. (2024) modeled compromise allocation as deterministic integer programming solved with intuitionistic fuzzy programming, showing via computations that it reduces variances and errors, improving precision in microeconomic surveys. Wesołowski et al. (2024) developed a recursive Neyman algorithm (RNABOX) for stratum sample sizes under box constraints, proving optimality with Karush-Kuhn-Tucker theory and implementing it in R as a generalization of classical Neyman allocation.

To obtain optimal or near-optimal solutions for multivariate sample allocation problems, general-purpose optimization techniques such as the Generalized Simulated Annealing Algorithm (Tsallis, 1996) and other metaheuristic methods can be applied, as demonstrated in this study.

Unlike prior work (e.g. Mahfouz et al., 2023) that developed specific stochastic models, our study introduces a broader comparison of stochastic, deterministic, and hybrid optimization algorithms under two canonical allocation formulations. We benchmark results against a globally optimal integer programming solution and apply the methods to both synthetic and real populations, including high-dimensional, skewed, and correlated data, providing new insights into practical performance under diverse conditions.

The structure of the paper is as follows. Section 2 outlines fundamental concepts and definitions related to stratified sampling and introduces two multivariate sample allocation problems along with relevant algorithms. Section 3 presents simulation results illustrating the performance of different allocation methods. Lastly, Section 4 offers concluding remarks.

2. Formulations and algorithms for sample allocation

Consider a finite population denoted by $\mathscr{U} = \{u_1, u_2, \ldots, u_N\}$, consisting of N distinct units. Assume there are m study variables $y^{(1)}, y^{(2)}, \ldots, y^{(m)}$, each defined over the population $\mathscr{U}$ and taking real values. For each variable $y^{(j)}$, the corresponding values for all population units are given by $y_1^{(j)}, y_2^{(j)}, \ldots, y_N^{(j)}$, where $j = 1, 2, \ldots, m$.

Now, suppose the population is partitioned into H non-overlapping and exhaustive strata, denoted by $\mathscr{U}_1, \mathscr{U}_2, \ldots, \mathscr{U}_H$, such that:

$$\mathscr{U} = \bigcup_{h=1}^{H} \mathscr{U}_h,$$

with each stratum $\mathscr{U}_h$ containing N_h units, for $h = 1,2,\ldots,H$. From each stratum, a simple random sample $\mathbf{s}_h \subset \mathscr{U}_h$ of size n_h is selected without replacement. The overall sample $\mathbf{s}$, the total population size N, and the total sample size n satisfy the following relationships:

$$\mathbf{s} = \bigcup_{h=1}^{H} \mathbf{s}_h, \quad N = \sum_{h=1}^{H} N_h, \quad n = \sum_{h=1}^{H} n_h.$$

The quantities of interest are the finite population totals for each variable:

$$t_j = \sum_{i=1}^{N} y_i^{(j)}, \quad j = 1,2,\ldots,m.$$

These totals, $t_1, t_2, \ldots, t_m$, can be estimated using the Horvitz-Thompson estimator (Horvitz and Thompson 1952):

$$\hat{t}_j = \sum_{h=1}^{H} \frac{N_h}{n_h} \sum_{i=1}^{n_h} y_{hi}^{(j)}, \quad j = 1,2,\ldots,m,$$

where $y_{hi}^{(j)}$ denotes the i-th observed value of variable $y^{(j)}$ in the sample $\mathbf{s}_h$ from stratum $\mathscr{U}_h$.

The variance of the estimator $\hat{t}_j$ for each $j = 1,2,\ldots,m$ is given by:

$$V(\hat{t}_j) = \sum_{h=1}^{H} N_h^2 \left(1 - \frac{n_h}{N_h}\right) \frac{s_{hj}^2}{n_h}, \tag{1}$$

where s_{hj}^2 denotes the variance of variable $y^{(j)}$ within stratum $\mathscr{U}_h$.

Because the variance (1) of the estimators is determined solely by the sample sizes chosen for each stratum – given that the number of strata (H), population sizes within strata (N_h), and within-stratum variances (s_{hj}^2 for $h = 1,2,\ldots,H$ and $j = 1,2,\ldots,m$) are fixed once the stratification is set – the level of variance can be managed through the appropriate selection of sample sizes $n_1, n_2, \ldots, n_H$. Consequently, decreasing these sample sizes leads to higher variance and a greater coefficient of variation in the total estimates, which in turn can reduce the accuracy of the results. Nonetheless, to limit the total survey cost, expressed as $\sum_{h=1}^{H} c_h n_h$, where c_h is the per-unit cost of sampling in stratum U_h, it is often necessary to reduce sample sizes. To address this trade-off, various sample allocation strategies – such as those developed by Kokan and Khan (1967), Bethel (1985, 1989), Ahsan and Khan (1982), Brito et al. (2015), among others – have been proposed to balance the need for precision in the survey variables of interest with cost efficiency.

Kish, L., (1976), Khan and Ahsan (2003), Garciá and Cortez (2006), Khan et al. (2011), and others have addressed the problem of optimal allocation in multivariate stratified sampling, focusing on optimizing allocation strategies with respect to the variances of estimators, under constraints such as total sample size or cost. Their work involves various mathematical programming approaches – including nonlinear, dynamic, and convex optimization – to balance precision and resource limitations in survey design.

The contributions discussed above allow us to distinguish two main directions in approaching the sample allocation problem. These perspectives form the basis for the two problem formulations presented below.

Problem 1. Find strata sample sizes $n_1, n_2, \ldots, n_H$, which minimize the total survey cost

$$C = \sum_{h=1}^{H} c_h n_h \tag{2}$$

and satisfy the following inequalities:

$$n_{\min} \leq n_h \leq N_h \quad (h = 1, \ldots, H), \tag{3}$$

$$\frac{\sqrt{V(\hat{t}_j)}}{t_j} \leq CV_j \quad (j = 1, \ldots, m), \tag{4}$$

where CV_j, for $j = 1, 2, \ldots, m$, are the pre-specified coefficients of variation of the estimators $\hat{t}_j$, $j = 1, 2, \ldots, m$.

In this formulation, constraint (3) ensures that each stratum receives a sample size between $n_{\min}$ and its population size. Constraint (4) keeps the coefficient of variation of each estimator within the target CV_j.

Problem 2. Find strata sample sizes $n_1, n_2, \ldots, n_H$ that minimize the weighted sum of the relative variances of the estimators of totals

$$\sum_{j=1}^{m} w_j \frac{1}{t_j^2} \sum_{h=1}^{H} s_{hj}^2 \frac{N_h^2}{n_h} \left(1 - \frac{n_h}{N_h}\right) \tag{5}$$

and satisfy the following inequalities:

$$n_{\min} \leq n_h \leq N_h \quad (h = 1, \ldots, H), \tag{6}$$

$$\sum_{h=1}^{H} c_h n_h \leq C^*, \tag{7}$$

where C^* is the total cost, defined as a function of the available survey budget. The weights w_j, for $j = 1, 2, \ldots, m$, are predetermined values associated with the importance of each variable of interest, such that $0 < w_j < 1$ and $w_1 + w_2 + \ldots + w_m = 1$.

The constraint in (6) is identical to that in (3), while the constraint in (7) ensures that the total cost is less than or equal to C^*, which is defined based on the available survey budget.

Bethel's Algorithm (BA). Bethel (1985, 1989) solved Problem 1 without incorporating constraint (3), and developed an algorithm that is guaranteed to converge to a solution (when one exists). This was achieved by applying the Kuhn and Tucker (1951) Theorem and the method of Lagrange multipliers to tackle the optimization problem. Later, some implementations – such as the `bethel()` function in the `SamplingStrata` package for the R programming language – modified the algorithm to incorporate constraint (3).

In this work, we also employ the **Generalized Simulated Annealing Algorithm (GSAA)** introduced by Tsallis and Stariolo (1996), designed for global optimization of real-valued

functions $C(\mathbf{n}) : \mathbb{R}^H \to \mathbb{R}$, where $C(\mathbf{n})$, in the context of our paper, corresponds to the survey cost function or a weighted sum of the relative variances of the estimators of totals, and $\mathbf{n} = (n_1, n_2, \ldots, n_H)$. While originally formulated for unconstrained problems, GSAA can be extended to constrained settings by incorporating penalty terms into the objective or by applying constraint-preserving sampling strategies.

The algorithm is based on a generalized entropy functional from nonextensive statistical mechanics:

$$S_q = k \frac{1 - \sum_i p_i^q}{q - 1}, \quad q \in \mathbb{R},$$

which reduces to the Shannon (1948) entropy as $q \to 1$. Here, $\{p_i\}$ denotes the probabilities of the microscopic configurations, and k is a conventional positive constant. The two main parameters of GSAA are q_V and q_A, which control the sampling distribution and the acceptance probability, respectively.

Candidate moves are generated using a power-law visiting distribution $g_{q_V}(\Delta \mathbf{n}_t)$, controlled by the parameter q_V, and the annealing temperature $T_{q_V}^{(V)}(t)$ at iteration t.

The acceptance of uphill moves is governed by a generalized Metropolis rule:

$$P_{q_A}(\mathbf{n}_t \to \mathbf{n}_{t+1}) = \begin{cases} 1 & \text{if } C(\mathbf{n}_{t+1}) < C(\mathbf{n}_t), \\ \left(1 + (q_A - 1)\dfrac{C(\mathbf{n}_{t+1}) - C(\mathbf{n}_t)}{T_{q_A}^{(A)}(t)}\right)^{\frac{1}{1-q_A}} & \text{otherwise,} \end{cases}$$

where $T_{q_A}^{(A)}(t)$ is the acceptance temperature at iteration t.

The temperature follows a generalized cooling schedule:

$$T_{q_V}^{(V)}(t) = T_{q_V}(1) \frac{2^{q_V - 1} - 1}{(1+t)^{q_V - 1} - 1},$$

ensuring slow enough cooling to maintain ergodicity and eventual convergence to the global minimum.

Thus, at each iteration t, a candidate solution $\mathbf{n}_{t+1}$ is proposed by drawing a displacement $\Delta \mathbf{n}_t$ from the visiting distribution $g_{q_V}(\Delta \mathbf{n}_t)$ centered at the current solution $\mathbf{n}_t$. The candidate is accepted with probability $P_{q_A}(\mathbf{n}_t \to \mathbf{n}_{t+1})$, and the relevant temperatures are updated according to their respective cooling schedules. The process repeats until convergence criteria are met, typically when the energy stabilizes or the maximum number of iterations is reached.

For constrained problems, the objective may be modified as $C^*(\mathbf{n}) = C(\mathbf{n}) + a \cdot \text{Penalty}(\mathbf{n})$, where $a > 0$ controls the penalty strength.

Thus, GSAA offers a powerful and flexible global optimization method, especially suitable for complex landscapes and adaptable to both unconstrained and constrained scenarios.

In derivative-free optimization with nonlinear constraints, Powell (1994) introduced **Constrained Optimization by Linear Approximations (COBYLA)**. The method builds local linear models of the objective $C(\mathbf{n}) : \mathbb{R}^H \to \mathbb{R}$ and constraints $G_i(\mathbf{n}) : \mathbb{R}^H \to \mathbb{R}$, $i = 1, \ldots, r$, by interpolating their values at the vertices of a non-degenerate H-simplex. From

a simplex $\{x^{(0)}, x^{(1)}, \ldots, x^{(H)}\} \subset \mathbb{R}^H$ with full affine span, linear approximations $\tilde{C}(\mathbf{n})$ and $\tilde{G}_i(\mathbf{n})$ are constructed to match the true functions at each vertex.

At each iteration, COBYLA solves a linear subproblem

$$\begin{aligned} &\min_{\mathbf{n}\in\mathbb{R}^H} \tilde{C}(\mathbf{n}) \\ &\text{subject to} \quad \tilde{G}_i(\mathbf{n}) \geq 0, \quad i = 1, \ldots, r, \\ &\qquad\qquad \|\mathbf{n}_{t+1} - \mathbf{n}_t\|_2 \leq \Delta_t, \end{aligned}$$

where $\mathbf{n}_t$ is the current best vertex (minimizing a merit function) and $\Delta_t > 0$ is the trust-region radius. The radius is reduced if sufficient merit decrease is not achieved, regardless of simplex geometry.

The merit function used to compare candidate points is defined as

$$\Phi(\mathbf{n}) = C(\mathbf{n}) + \mu \cdot \max_{1\leq i\leq r} [-G_i(\mathbf{n})]_+ ,$$

with penalty parameter $\mu > 0$ dynamically updated to balance objective and constraint satisfaction. If the linearized subproblem is infeasible, COBYLA minimizes maximum constraint violation under the trust region. The simplex is then updated either by incorporating a new feasible point or by improving the interpolation geometry.

Although theoretical convergence guarantees are limited, COBYLA performs well in practice for low-dimensional problems without reliable derivatives, making it useful for black-box or noisy applications in engineering and science.

Consider the optimization of a real-valued objective function $C(\mathbf{n}) : \mathbb{R}^H \to \mathbb{R}$, defined over a bounded search domain $\Omega \subset \mathbb{R}^H$, where the goal is to find $\mathbf{n}^* \in \Omega$ such that $C(\mathbf{n}^*) = \min_{\mathbf{n}\in\Omega} C(\mathbf{n})$. To solve this, we use the **Particle Swarm Optimization Algorithm (PSOA)** (Kennedy and Eberhart 1995), which models a swarm of particles sharing positional information to locate the global minimum. Each particle i has a position $\mathbf{n}_{i,t} \in \Omega$, velocity $\mathbf{v}_{i,t}$, personal best position $\mathbf{p}_{i,t}$, and neighborhood best position $\mathbf{l}_{i,t}$. Velocities evolve as

$$\mathbf{v}_{i,t+1} = F(\mathbf{v}_{i,t}, \mathbf{p}_{i,t} - \mathbf{n}_{i,t}, \mathbf{I}_{i,t} - \mathbf{n}_{i,t}),$$

and positions update iteratively by

$$\mathbf{n}_{i,t+1} = \mathbf{n}_{i,t} + \mathbf{v}_{i,t+1},$$

with $C(\mathbf{n}_{i,t})$ guiding updates of $\mathbf{p}_{i,t}$ and $\mathbf{l}_{i,t}$.

To ensure convergence, Clerc and Kennedy (2002) introduced a constriction factor χ, yielding the standard PSOA update:

$$\mathbf{v}_{i,t+1} = \chi\left(\mathbf{v}_{i,t} + \alpha_1 \boldsymbol{\beta}_1 (\mathbf{p}_{i,t} - \mathbf{n}_{i,t}) + \alpha_2 \boldsymbol{\beta}_2 (\mathbf{l}_{i,t} - \mathbf{n}_{i,t})\right),$$

where α_1, α_2 are acceleration coefficients, and $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2$ are vectors of independent random samples drawn from the uniform distribution $U(0,1)$. Clerc (2012) later formalized Standard PSOA versions with reproducibility, rigorous topologies, and boundary handling.

Further developments include the phasor PSOA (PPSOA) (Ghasemi et al. 2018), which uses trigonometric phase-based updates for parameter-free adaptivity, and the multi-phase PSOA (Li et al. 2021), which segments optimization into phases with distinct strategies for improved performance.

For constrained optimization, modified PSOA methods incorporate strategies such as penalty functions, feasibility rules, and repair operators to enforce constraints while preserving swarm behavior (Rini et al. 2011).

PSOA has broad applicability. In statistical sampling, it optimizes stratum boundaries to minimize estimator variance under Neyman allocation (Al-Kassab and Ali 2015). In power systems, it is widely used for economic dispatch, optimal power flow, and reactive power control (del Valle et al. 2008). Numerous enhancements – such as inertia weight schedules, topology control, and hybridization with mutation operators – have been surveyed by Imran et al. (2013), underscoring the algorithm's adaptability and continued development.

In all cases, the central aim remains to iteratively improve candidate solutions $\mathbf{n}_{i,t}$ such that $C(\mathbf{n}_{i,t})$ approaches the global minimum, exploiting both individual and collective experience within the swarm framework.

In this study, we utilize the **Biased Random-Key Genetic Algorithm (BRKGA)** (Gonçalves and Resende 2011), an extension of the original Random-Key Genetic Algorithm (RKGA) proposed by Bean (1994), which can be applied to optimize a real-valued objective function $C(\mathbf{n}) : \mathbb{R}^H \to \mathbb{R}$. In the RKGA, candidate solutions are encoded as chromosomes – vectors of real numbers drawn from the interval $[0,1]$ – which are decoded into feasible solutions using a problem-specific mapping. This indirect encoding offers flexibility and is well-suited for combinatorial optimization problems.

The BRKGA, as applied by Brito et al. (2022), modifies the standard RKGA by introducing biased selection during crossover. In each generation, a population of N^* chromosomes is divided into an elite set (best-performing solutions), a non-elite set, and a set of mutant chromosomes randomly generated to preserve diversity. Crossover is performed between pairs where one parent is always selected from the elite set and the other from the non-elite set. A uniformly random auxiliary vector $\mathbf{v}_{aux} \in [0,1]^H$ and a predefined bias parameter $\delta_e > 0.5$ guide gene inheritance: if $v_{aux,i} \leq \delta_e$, the offspring gene at position i is inherited from the elite parent; otherwise, from the non-elite.

Each chromosome $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_H)$, where H is the number of strata, is decoded into a vector $\boldsymbol{\nu} = (n_1, \ldots, n_H)$ of sample sizes via a decoder. For the formulation that minimizes the total survey cost under precision constraints (Problem 1), assuming unit costs $c_h = 1$ for all strata, $h = 1, \ldots, H$, the decoding is given by:

$$n_h = n_{\min} + \operatorname{round}\left(\gamma_h \cdot (N_h - n_{\min})\right),$$

ensuring that $n_h \in [n_{\min}, N_h]$, $h = 1, \ldots, H$. For the formulation that minimizes a weighted sum of relative variances under a fixed total survey cost (Problem 2) – which reduces to the total sample size n when $c_h = 1$ for all $h = 1, \ldots, H$ – the decoding follows:

$$n_h = n_{\min} + \left((n - H n_{\min}) \cdot \frac{\gamma_h}{\sum_{k=1}^{H} \gamma_k} \right) \quad \text{for } h = 1, \ldots, H-1, \quad n_H = n - \sum_{h=1}^{H-1} n_h.$$

After decoding, each solution is evaluated using the objective function. To enforce feasibility, a penalty term is added if any constraint is violated. For instance, in Problem 1, the penalized objective becomes:

$$C_p = \sum_{h=1}^{H} n_h + P,$$

where $P = T^M$, $T \in \mathbb{R}$, if any $CV(\hat{t}_y^{(j)}) > CV_j$, and zero otherwise. Here, $M = \max_j \left\{ \frac{CV(\hat{t}_y^{(j)})}{CV_j} \right\}$.

This procedure ensures that only feasible or near-feasible solutions persist across generations. The BRKGA evolves the population by keeping elites, adding mutants, and producing biased offspring, balancing exploration and exploitation. As shown by Brito et al. (2022), it yields high-quality integer-feasible solutions under nonlinear constraints, rivaling exact integer programming methods.

Most approaches used to derive optimal sample sizes face challenges related to rounding, which can be particularly problematic in certain scenarios. These include: (1) surveys involving small areas, where adding or removing even a single unit from the sample can notably affect the variance estimates, and (2) surveys with a very high number of strata, where the total sample size n may differ considerably from the sum of the individually rounded sample sizes allocated to each stratum.

To address these issues, Brito et al. (2015) proposed **Integer Programming Algorithms (IPA)** to solve Problems 1 and 2, with the following additional constraint imposed: $n_h \in \mathbb{Z}_+$, for $h = 1, \ldots, H$. They used simple algebraic techniques to achieve linearity either in the objective function or in the constraints. Specifically, Brito et al. (2015) introduced a new binary variable z defined as:

$$z_{hk} = \begin{cases} 1, & \text{if the sample size } k \in \{n_{min}, \ldots, N_h\}, h = 1, \ldots, H, \text{ is allocated to stratum } U_h; \\ 0, & \text{otherwise.} \end{cases}$$

Through this variable, the second constraint in Problem 1 – which is originally nonlinear – can be reformulated as a linear expression in terms of the values of the binary variable z:

$$\sum_{h=1}^{H} N_h p_{hj} \sum_{k=n_{min}}^{N_h} \frac{z_{hk}}{k} - \sum_{h=1}^{H} p_{hj} \leq 1, \quad p_{hj} = \frac{N_h s_{hj}^2}{t_j^2 CV_j^2}, \quad j = 1, 2, \ldots, m.$$

Similarly, the nonlinear objective function in Problem 2 can be reformulated as a linear expression in terms of the binary variable z:

$$\sum_{j=1}^{m} w_j \frac{1}{t_j^2} \sum_{h=1}^{H} \left(\sum_{k=n_{\min}}^{N_h} \frac{z_{hk}}{k} \right) N_h^2 s_{hj}^2.$$

Although not shown here, the remaining constraints in Problems 1 and 2, as well as the linear objective function in Problem 1, are also reformulated in terms of the binary variable z, resulting in fully linear expressions.

After achieving linearity either in the objective function or in the constraints, Brito et al. (2015) solved the resulting integer programming problems using the Branch and Bound method (Wolsey 1998). This optimization approach guarantees attainment of the global minimum.

3. Numerical comparisons

This section presents the findings from a comparison of various multivariate optimal allocation techniques applied to a specific subset of population datasets. All computations were performed using the R programming language. The evaluation focuses on several algorithms employed to solve Problem 1, including Integer Programming (IPA) (Brito et al. 2015a), Bethel's Algorithm (BA) (Bethel 1985, 1989), the Generalized Simulated Annealing Algorithm (GSAA) (Tsallis and Stariolo 1996), Constrained Optimization by Linear Approximations (COBYLA) (Powell 1994), Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995), and the Biased Random Key Genetic Algorithm (BRKGA) (Gonçalves and Resende 2011; Brito et al. 2022). The IPA, GSAA, COBYLA, PSOA, and BRKGA algorithms are also applied to solve Problem 2, along with the textbook method given in Cochran (1977), which is denoted as TBA. Specifically, according to this method, the optimal sample size n_h from stratum $\mathscr{U}_h$ is calculated using the following formula:

$$n_h = n\frac{\sqrt{\sum_{j=1}^{m}\left(n_{hj}^{(N)}\right)^2}}{\sum_{h=1}^{H}\sqrt{\sum_{j=1}^{m}\left(n_{hj}^{(N)}\right)^2}},$$

where $n_{hj}^{(N)}$ denotes the optimum sample size in stratum $\mathscr{U}_h$ for variable j, calculated according to the Neyman (1934) allocation.

Initially, the comparisons are performed on two synthetic populations, each containing $N = 10000$ units. In each population, four study variables are specified. To establish a predetermined dependence structure among them, a Gaussian copula is first constructed. This copula serves as a probability distribution where each of the four random variables has a uniform marginal distribution. Next, these uniformly distributed variables are converted into the target distributions by applying the inverse transform method.

Thus, for Population 1, the variables are simulated from asymmetric distributions: $y^{(1)} \sim \mathscr{E}(0.005)$, $y^{(2)} = |y|$, where $y \sim t(3)$, $y^{(3)} \sim \Gamma(1,2)$, $y^{(4)} \sim \chi^2(2)$, with $\rho(y^{(1)},y^{(2)}) = 0.13$, $\rho(y^{(1)},y^{(3)}) = 0.39$, $\rho(y^{(1)},y^{(4)}) = -0.31$, $\rho(y^{(2)},y^{(3)}) = 0.12$, $\rho(y^{(2)},y^{(4)}) = 0.13$, $\rho(y^{(3)},y^{(4)}) = 0.30$.

Population 2 consists of study variables following a combination of normal, exponential, and Fisher distributions: $y^{(1)} \sim \mathscr{E}(0.005)$, $y^{(2)} \sim \mathscr{N}(3000,300)$, $y^{(3)} \sim \mathscr{F}(5,4)$, and $y^{(4)} \sim \mathscr{N}(100,20)$. Here, the second parameter in the normal distributions represents the standard deviation. The correlations between these variables are given as follows: $\rho(y^{(1)},y^{(2)}) = 0.19$, $\rho(y^{(1)},y^{(3)}) = 0.13$, $\rho(y^{(1)},y^{(4)}) = 0.27$, $\rho(y^{(2)},y^{(3)}) = 0.12$, $\rho(y^{(2)},y^{(4)}) = 0.20$, $\rho(y^{(3)},y^{(4)}) = 0.17$.

For extended analysis, Population 3 is introduced, originating from a statistical survey on the area and yield of agricultural plants in Lithuanian agricultural companies and en-

terprises, with a total population size of $N = 6\,204$. To assess different sample allocation methods, the following four skewed variables are selected: $y^{(1)}$ - total yield of cereals and oilseed rape, $y^{(2)}$ - total yield of cereals and oilseed rape after cleaning and drying, $y^{(3)}$ - total area of cereals and oilseed rape, and $y^{(4)}$ - total harvested area of cereals and oilseed rape. The relationships between these variables are characterized by the following correlation coefficients: $\rho(y^{(1)}, y^{(2)}) = 0.64$, $\rho(y^{(1)}, y^{(3)}) = 0.66$, $\rho(y^{(1)}, y^{(4)}) = 0.71$, $\rho(y^{(2)}, y^{(3)}) = 0.86$, $\rho(y^{(2)}, y^{(4)}) = 0.87$, $\rho(y^{(3)}, y^{(4)}) = 0.93$.

All populations are stratified using the traditional k-means approach implemented in the `stats` package in R (R Core Team 2023). The number of strata, H, along with their respective sizes, $N_1, N_2, \ldots, N_H$, are detailed in Table 1.

Table 1. Population strata sizes and number of strata

Population	Number of strata (H)	Strata sizes ($N_1, N_2, \ldots, N_H$)
1	10	1024, 531, 501, 1253, 1761, 649, 1616, 1211, 731, 723
2	7	1083, 831, 245, 1397, 2719, 1123, 2602
3	6	680, 438, 557, 695, 2596, 1238

For the numerical analysis conducted in each population, unit survey costs are assumed to be the same across all strata, and the weights w_j, for $j = 1, 2, \ldots, m$, are considered equal for all survey variables. A minimum sample size per stratum of $n_{\min} = 2$ is maintained for all methods and populations examined. The predefined coefficients of variation for the total estimators in Problem 1 are set at 5%, 10%, and 15%. In Problem 2, the total sample sizes for allocation are determined based on sampling fractions of 5%, 10%, and 20% of the respective population sizes (N).

The algorithms are implemented using their respective R packages, with specific parameter configurations as detailed below. The Integer Programming Algorithms (IPA) employ the functions `BSSM_FC()` and `BSSM_FD()` from the `MultAlloc` package (Brito et al. 2015b) to solve Problem 1 and Problem 2, respectively. Bethel's Algorithm (BA) is executed via the `bethel()` function from the `SamplingStrata` package (Barcaroli 2014), where precision constraints are defined in terms of coefficients of variation for each studied variable. The textbook method (TBA) is developed by us using the R programming language. For the generalized simulated annealing algorithm (GSAA), the `GenSA()` function from the `GenSA` package (Xiang et al. 2013) is used, with the parameters set as follows: `temperature = 1000`, `parameter for visiting distribution = 2.63`, and `parameter for acceptance distribution = -12`. Constrained Optimization by Linear Approximations (COBYLA) employs the `nloptr()` function from the `nloptr` package, with the `algorithm` parameter assigned the value `NLOPT_LN_COBYLA`. Particle Swarm Optimization (PSO) is carried out using the `psoptim()` function from the `pso` package (Bendtsen 2022), where the `swarm size` is configured as 300. By default, this algorithm adheres to the Standard PSOA 2007 framework established by Clerc (2012). Lastly, the Biased Random Key Genetic Algorithm (BRKGA) is executed using the `brkga()` function from the `BRKGA` package (Brito et al. 2023), with the following parameter settings: `size of the algorithm population = 2000`, `percentage of elite chromosomes = 0.2`, `percentage of mutant chromosomes = 0.2`, `crossover probability = 0.6`,

`number of generations = 2 000`, and `penalty factor = 1 000`. Any other hyperparameters that are not explicitly stated remain at their default values. The R code defining the objective function and constraints for GSAA, COBYLA, PSOA, and BRKGA is created externally from their respective function environments.

Table 2. Comparison of Algorithms for Population 1 and Problem 1

Alg.:	IPA	BA	GSAA	CBLA	PSOA	BRKGA
$CV_j = 5\%$						
$\sum n_i$	**457**	462	458	460	460	**457**
$f(\text{in }\%)$	4.570	4.620	4.580	4.600	4.600	4.570
$CV(\hat{t}_y^{(1)})$	3.408	3.386	3.389	3.389	3.418	3.401
$CV(\hat{t}_y^{(2)})$	4.996	4.967	4.999	4.978	4.997	4.999
$CV(\hat{t}_y^{(3)})$	4.550	4.522	4.519	4.537	4.582	4.581
$CV(\hat{t}_y^{(4)})$	4.485	4.459	4.479	4.472	4.504	4.492
$CV_j = 10\%$						
$\sum n_i$	**119**	124	**119**	**119**	120	120
$f(\text{in }\%)$	1.190	1.240	1.190	1.190	1.200	1.200
$CV(\hat{t}_y^{(1)})$	6.809	6.695	6.778	6.809	6.985	6.702

Alg.:	IPA	BA	GSAA	CBLA	PSOA	BRKGA
$CV(\hat{t}_y^{(2)})$	9.983	9.777	9.999	9.983	9.998	9.980
$CV(\hat{t}_y^{(3)})$	9.047	8.896	9.049	9.047	9.185	8.953
$CV(\hat{t}_y^{(4)})$	8.948	8.750	8.948	8.948	8.921	8.959
$CV_j = 15\%$						
$\sum n_i$	**54**	59	**54**	**54**	**54**	**54**
$f(\text{in }\%)$	0.540	0.590	0.540	0.540	0.540	0.540
$CV(\hat{t}_y^{(1)})$	10.313	9.786	10.214	10.306	10.214	10.070
$CV(\hat{t}_y^{(2)})$	14.937	14.269	14.974	14.889	14.968	14.987
$CV(\hat{t}_y^{(3)})$	13.636	12.831	13.520	13.576	13.432	13.495
$CV(\hat{t}_y^{(4)})$	13.418	12.750	13.388	13.276	13.353	13.447

Table 3. Comparison of Algorithms for Population 2 and Problem 1

Alg.:	IPA	BA	GSAA	CBLA	PSOA	BRKGA
$CV_j = 5\%$						
$\sum n_i$	**773**	776	**773**	774	774	**773**
$f(\text{in }\%)$	7.730	7.760	7.730	7.740	7.740	7.730
$CV(\hat{t}_y^{(1)})$	2.497	2.473	2.459	2.495	2.455	2.478
$CV(\hat{t}_y^{(2)})$	0.192	0.190	0.189	0.192	0.189	0.190
$CV(\hat{t}_y^{(3)})$	4.998	4.982	4.999	4.992	4.995	5.000
$CV(\hat{t}_y^{(4)})$	1.002	0.994	0.995	1.001	0.983	0.995
$CV_j = 10\%$						
$\sum n_i$	**305**	308	**305**	306	306	**305**
$f(\text{in }\%)$	3.050	3.080	3.050	3.060	3.060	3.050
$CV(\hat{t}_y^{(1)})$	4.266	4.140	4.166	4.264	4.144	4.127

Alg.:	IPA	BA	GSAA	CBLA	PSOA	BRKGA
$CV(\hat{t}_y^{(2)})$	0.328	0.318	0.319	0.328	0.318	0.317
$CV(\hat{t}_y^{(3)})$	9.988	9.928	9.999	9.967	9.997	9.999
$CV(\hat{t}_y^{(4)})$	1.710	1.665	1.681	1.709	1.648	1.655
$CV_j = 15\%$						
$\sum n_i$	**153**	156	**153**	155	**153**	154
$f(\text{in }\%)$	1.530	1.560	1.530	1.550	1.530	1.540
$CV(\hat{t}_y^{(1)})$	6.042	5.714	5.773	6.038	5.903	5.702
$CV(\hat{t}_y^{(2)})$	0.465	0.438	0.442	0.464	0.451	0.436
$CV(\hat{t}_y^{(3)})$	14.976	14.823	14.993	14.868	14.998	14.993
$CV(\hat{t}_y^{(4)})$	2.419	2.301	2.347	2.418	2.402	2.303

Table 4. Comparison of Algorithms for Population 3 and Problem 1

Alg.:	IPA	BA	GSAA	CBLA	PSOA	BRKGA
$CV_j = 5\%$						
$\sum n_i$	**745**	747	**745**	747	**745**	**745**
$f(\text{in }\%)$	12.008	12.041	12.008	12.041	12.008	12.008
$CV(\hat{t}_y^{(1)})$	4.996	4.987	4.999	4.987	4.998	5.000
$CV(\hat{t}_y^{(2)})$	3.758	3.748	3.771	3.754	3.817	3.735
$CV(\hat{t}_y^{(3)})$	4.242	4.238	4.293	4.240	4.400	4.077
$CV(\hat{t}_y^{(4)})$	3.521	3.516	3.579	3.519	3.649	3.427
$CV_j = 10\%$						
$\sum n_i$	**229**	231	**229**	**229**	**229**	**229**
$f(\text{in }\%)$	3.691	3.723	3.691	3.691	3.691	3.691
$CV(\hat{t}_y^{(1)})$	9.984	9.936	10.000	9.976	9.995	9.982

Alg.:	IPA	BA	GSAA	CBLA	PSOA	BRKGA
$CV(\hat{t}_y^{(2)})$	7.390	7.271	7.337	7.407	7.587	7.309
$CV(\hat{t}_y^{(3)})$	7.455	7.592	7.605	7.821	8.443	7.790
$CV(\hat{t}_y^{(4)})$	6.421	6.351	6.376	6.618	6.997	6.575
$CV_j = 15\%$						
$\sum n_i$	**107**	110	**107**	**107**	**107**	**107**
$f(\text{in }\%)$	1.725	1.773	1.725	1.725	1.725	1.725
$CV(\hat{t}_y^{(1)})$	14.953	14.735	14.989	14.953	14.982	14.992
$CV(\hat{t}_y^{(2)})$	10.795	10.649	10.825	10.795	10.848	10.824
$CV(\hat{t}_y^{(3)})$	10.858	10.797	10.908	10.858	10.874	10.001
$CV(\hat{t}_y^{(4)})$	9.100	9.028	9.130	9.100	9.121	8.656

Table 5. Comparison of Algorithms for Population 1 and Problem 2

Alg.:	IPA	TBA	GSAA	CBLA	PSOA	BRKGA
$n = 500$, $f = 5\%$						
$CV(\hat{t}_y^{(1)})$	3.228	3.603	3.247	3.238	3.205	3.238
$CV(\hat{t}_y^{(2)})$	4.787	5.281	4.788	4.789	4.806	4.789
$CV(\hat{t}_y^{(3)})$	4.255	4.581	4.272	4.254	4.292	4.254
$CV(\hat{t}_y^{(4)})$	4.275	4.532	4.250	4.266	4.318	4.266
$\sum CV(\hat{t}_y^{(i)})$	**16.545**	17.997	16.557	16.547	16.621	16.547
$n = 1000$, $f = 10\%$						
$CV(\hat{t}_y^{(1)})$	2.206	2.479	2.212	2.214	2.211	2.209
$CV(\hat{t}_y^{(2)})$	3.287	3.635	3.290	3.290	3.287	3.287
$CV(\hat{t}_y^{(3)})$	2.933	3.164	2.944	2.933	2.929	2.929
$CV(\hat{t}_y^{(4)})$	2.940	3.122	2.931	2.942	2.940	2.941
$\sum CV(\hat{t}_y^{(i)})$	**11.366**	12.400	11.377	11.379	11.367	**11.366**
$n = 2000$, $f = 20\%$						
$CV(\hat{t}_y^{(1)})$	1.455	1.654	1.465	1.453	1.453	1.453
$CV(\hat{t}_y^{(2)})$	2.179	2.437	2.184	2.177	2.177	2.178
$CV(\hat{t}_y^{(3)})$	1.950	2.124	1.970	1.954	1.954	1.953
$CV(\hat{t}_y^{(4)})$	1.961	2.096	1.957	1.961	1.961	1.961
$\sum CV(\hat{t}_y^{(i)})$	**7.545**	8.311	7.576	**7.545**	**7.545**	**7.545**

Table 6. Comparison of Algorithms for Population 2 and Problem 2

Alg.:	IPA	TBA	GSAA	CBLA	PSOA	BRKGA
$n = 500$, $f = 5\%$						
$CV(\hat{t}_y^{(1)})$	2.919	2.362	2.926	2.726	2.920	2.914
$CV(\hat{t}_y^{(2)})$	0.222	0.187	0.223	0.208	0.223	0.222
$CV(\hat{t}_y^{(3)})$	7.199	21.043	7.198	7.499	7.198	7.201
$CV(\hat{t}_y^{(4)})$	1.189	0.859	1.190	1.113	1.189	1.194
$\sum CV(\hat{t}_y^{(i)})$	**11.529**	24.451	11.537	11.546	11.530	11.531
$n = 1000$, $f = 10\%$						
$CV(\hat{t}_y^{(1)})$	1.847	1.627	1.859	1.854	1.854	1.847
$CV(\hat{t}_y^{(2)})$	0.140	0.129	0.142	0.141	0.141	0.141
$CV(\hat{t}_y^{(3)})$	4.142	14.438	4.136	4.137	4.137	4.141
$CV(\hat{t}_y^{(4)})$	0.755	0.591	0.759	0.758	0.758	0.755
$\sum CV(\hat{t}_y^{(i)})$	**6.884**	16.785	6.896	6.890	6.890	**6.884**
$n = 2000$, $f = 20\%$						
$CV(\hat{t}_y^{(1)})$	1.155	1.093	1.184	1.162	1.162	1.161
$CV(\hat{t}_y^{(2)})$	0.087	0.087	0.090	0.088	0.088	0.088
$CV(\hat{t}_y^{(3)})$	2.463	9.475	2.453	2.458	2.458	2.458
$CV(\hat{t}_y^{(4)})$	0.472	0.396	0.480	0.476	0.476	0.475
$\sum CV(\hat{t}_y^{(i)})$	**4.177**	11.051	4.207	4.184	4.184	4.182

Table 7. Comparison of Algorithms for Population 3 and Problem 2

Alg.:	IPA	TBA	GSAA	CBLA	PSOA	BRKGA
$n = 310$, $f = 5\%$						
$CV(\hat{t}_y^{(1)})$	8.825	9.895	8.881	8.844	8.825	8.825
$CV(\hat{t}_y^{(2)})$	5.843	7.218	5.852	5.827	5.843	5.843
$CV(\hat{t}_y^{(3)})$	4.963	5.225	4.937	4.984	4.963	4.963
$CV(\hat{t}_y^{(4)})$	4.392	4.966	4.359	4.409	4.392	4.392
$\sum CV(\hat{t}_y^{(i)})$	**24.023**	27.304	24.029	24.064	**24.023**	**24.023**
$n = 620$, $f = 10\%$						
$CV(\hat{t}_y^{(1)})$	5.906	6.697	5.905	5.940	5.930	5.904
$CV(\hat{t}_y^{(2)})$	3.888	4.896	3.884	3.948	3.901	3.889
$CV(\hat{t}_y^{(3)})$	3.376	3.556	3.382	3.363	3.359	3.377
$CV(\hat{t}_y^{(4)})$	2.970	3.379	2.976	2.976	2.950	2.971
$\sum CV(\hat{t}_y^{(i)})$	**16.140**	18.528	16.147	16.227	**16.140**	16.141
$n = 1241$, $f = 20\%$						
$CV(\hat{t}_y^{(1)})$	3.657	4.292	3.638	3.657	3.657	3.657
$CV(\hat{t}_y^{(2)})$	2.377	3.182	2.392	2.378	2.378	2.378
$CV(\hat{t}_y^{(3)})$	2.177	2.323	2.189	2.177	2.177	2.177
$CV(\hat{t}_y^{(4)})$	1.886	2.209	1.895	1.886	1.886	1.886
$\sum CV(\hat{t}_y^{(i)})$	**10.097**	12.006	10.114	10.098	10.098	10.098

Tables 2-4 present the sample allocation resulting from the solution to Problem 1, along with the total sample size, sampling fraction f, and the achieved coefficients of variation for the estimators of all variables across the populations, as well as the pre-specified coefficients of variation. Note that COBYLA is abbreviated as CBLA in the tables to ensure a better fit within the table format.

Among all the methods examined, only the Integer Programming Algorithm (IPA) attains the global minimum. Thus, other methods can be compared to IPA's results to evaluate deviations from the minimum objective value. Bolded values in the tables indicate the

global optimum, making it easier to visually compare the results of IPA with those of the other algorithms.

As shown in the tables, the total sample size n produced by Bethel's Algorithm (BA) is never smaller than that obtained by the other methods. The only case where BA matches another method is in Population 3 for $CV_j = 5\%$, where it yields the same total sample size as COBYLA. The greatest deviations from the global minimum for BA are observed in the skewed Population 1. Notably, BA does not achieve the global minimum in any of the analyzed populations.

Tables 2-4 also show that the algorithms GSAA, COBYLA, PSOA, and BRKGA achieve the global minimum in 88.89%, 44.44%, 55.56%, and 77.78% of the cases, respectively. In the remaining cases, the differences from the global minimum for these methods are not substantial. The lowest percentage of global minimum attainment, when GSAA, COBYLA, PSOA, and BRKGA are considered together, is observed in Population 2. In real Population 3, the methods discussed in this paragraph reach the global minimum in nearly all cases. The proportion of cases in which the global minimum is reached in Population 1 is similar to that in Population 2. We further observe that even if an algorithm finds the global minimum, the specific solution it yields may differ from that produced by IPA. As an example, in Population 2 where $CV_j = 10\%$, IPA produces the values $n_1, n_2, \ldots, n_H$ as 24, 42, 118, 8, 43, 48, and 22. Although GSAA also reaches the global minimum, it results in a different allocation: 24, 46, 113, 9, 42, 49, and 22. It is also worth noting that a method capable of achieving the global minimum for a fixed population and a particular value of CV_j may not consistently reach the global minimum when CV_j varies. As an illustration, in Population 2, PSOA successfully attains the global minimum at $CV_j = 15\%$, yet it does not achieve this outcome when CV_j is set to 5% or 10%.

The comparison of artificial Populations 1 and 2 indicates that both stratum and total sample sizes are sensitive to the distributional characteristics of the study variables. In Population 2, despite two variables being normally distributed, the presence of two skewed variables – particularly the one following a heavily skewed Fisher distribution – leads to a notable increase in the total sample size. Compared to other populations, real Population 3 demonstrates the most pronounced increase in sampling fraction for every pre-specified coefficient of variation.

Tables 5–7 present the comparative results of six algorithms – IPA, TBA, GSAA, COBYLA, PSOA, and BRKGA – for the previously considered populations under Problem 2. The objective in this setting is to minimize the weighted relative variance of the Horvitz-Thompson estimators of totals across multiple survey variables, given fixed overall sample sizes corresponding to sampling fractions of 5%, 10%, and 20%. Each table presents the sample allocations across strata, the corresponding coefficients of variation for each survey variable, and the total sum of coefficients of variation, $\sum CV(\hat{t}_y^{(i)})$, which serves as a summary measure of overall efficiency across all survey variables. The global minimum of this total is indicated in bold.

In these experiments, IPA is again used as a benchmark, as it achieves the global minimum in every case. Other methods are evaluated against IPA's performance in terms of both efficiency and allocation stability.

Thus, IPA serves as the reference algorithm, consistently achieving the lowest possible value of $\sum CV(\hat{t}_y^{(i)})$ across all populations and for each fixed total sample size. Its allocations are well-balanced and establish the best-case baseline against which all other methods are compared. BRKGA closely matches IPA in all settings, often reaching the same total *CV* values and producing well-balanced and robust allocations. It proves to be a competitive and stable alternative to IPA. COBYLA performs slightly worse than BRKGA, sometimes matching IPA and BRKGA in total *CV*. It provides consistent and efficient allocations, especially in Populations 1 and 3, making it quite a reliable method in practice. GSAA delivers results similar to COBYLA and BRKGA in some instances but shows occasional variability in allocation that slightly affects performance. PSOA performs moderately well across all populations, with results typically falling slightly above the optimal $\sum CV(\hat{t}_y^{(i)})$ values. Its allocations are generally balanced, although not as consistently efficient as IPA or BRKGA. TBA, while effective in select settings, often exhibits unstable behavior, particularly in Population 2. It tends to heavily over- or under-sample certain strata, which leads to significantly inflated coefficients of variation for some estimators (e.g. estimator $\hat{t}_y^{(3)}$). These outliers frequently result in high total *CV* values, thereby undermining the method's overall reliability.

4. Conclusions

The study finds that the Integer Programming Algorithm (IPA) is the most robust and accurate method for multivariate optimal allocation in stratified sampling. As an exact approach, it guarantees the global minimum, directly handles integer constraints, and avoids rounding errors – crucial for many strata or small-area surveys.

Bethel's Algorithm (BA), while widely used, consistently underperforms. It never reaches the global minimum and often yields the largest sample sizes, particularly struggling in populations with skewed distributions. Despite typically converging, Bethel's Algorithm often produces suboptimal results compared to more advanced methods.

Among stochastic approaches, the Biased Random-Key Genetic Algorithm (BRKGA) proves to be the most competitive. It frequently finds solutions that match or closely approximate IPA's and consistently delivers stable, well-balanced allocations. Its efficiency and adaptability make it a strong practical alternative, especially in scenarios where flexibility or faster approximate solutions are preferred over exact methods. The Generalized Simulated Annealing Algorithm (GSAA) also performs well, often reaching the global minimum, although with slightly more variability. COBYLA is reliable, especially with real-world data in Problem 1, although less consistent in achieving optimality in Problem 2. Particle Swarm Optimization (PSOA) offers reasonable results but tends to be more variable and less efficient under tighter precision constraints in certain cases.

The textbook method (TBA) is unstable in variance minimization, often yielding poor allocation balance and high coefficients of variation in some strata and variables.

The study shows that variable distribution affects allocation: skewed variables require larger samples in Problem 1 and yield higher variation in Problem 2, stressing the need for adaptive, precise methods.

In the real population case (Population 3), under Problem 1, for each predefined precision level ($CV_j = 5\%, 10\%$, and 15%), the sampling fraction is consistently higher than in the artificial populations, highlighting the greater complexity and variability inherent in real-world data. In Problem 2, which aims to minimize overall variance given a fixed total sample size, the highest coefficients of variation across all survey variables are also observed in the real data case.

This increase – whether in the sampling fraction under Problem 1 or in the coefficients of variation under Problem 2 – is primarily driven by the skewed distribution of the study variables, which necessitates larger samples to meet precision requirements in Problem 1 and leads to higher variances in Problem 2.

This study offers a comparative analysis of exact and approximate optimization methods. By benchmarking against exact integer programming and testing on diverse real and synthetic populations, it stands as one of the most comprehensive empirical studies of multivariate stratified sampling allocation. These results not only affirm the strengths and limitations of different algorithm classes but also provide actionable guidance for practitioners and survey designers facing real-world complexity in cost and variance trade-offs.

Acknowledgment

I sincerely thank Vilma Nekrašaitė-Liegė (VILNIUS TECH) for sharing the real-world dataset used in this study.

References

Ahsan, M. J., Khan, S. U., (1982). Optimum allocation in multivariate stratified random sampling with overhead cost. *Metrika*, 29, pp. 71–78. Available from: https://doi.org/10.1007/BF01893366.

AL-Kassab, M. M., Ali, A. A., (2015). Using particle swarm optimization to determine the optimal strata boundaries. *J. Adv. Math.*, 11(1). Available from: https://rajpub.com/index.php/jam/article/view/1290.

Barcaroli, G., (2014). SamplingStrata: An R package for the optimization of stratified sampling. *J. Stat. Softw.*, 61(4), pp. 1–24. Available from: https://doi.org/10.18637/jss.v061.i04.

Bean, J. C., (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA J. Comput.*, 6(2), pp. 154–160. Available from: https://doi.org/10.1287/ijoc.6.2.154.

Bendtsen, C., (2022). *pso: Particle Swarm Optimization*. Available from: https://cran.r-project.org/web/packages/pso/index.html. R package version 1.0.4.

Bethel, J., (1985). An optimum allocation algorithm for multivariate surveys. *Proc. Surv. Res. Methods Sect.*, pp. 209–212. Available from: http://www.asasrms.org/Proceedings/papers/$1985_0$35.pdf.

Bethel, J., (1989). Sample allocation in multivariate surveys. *Surv. Methodol.*, 15(1), pp. 47–57. Available from: https://www.istat.it/en/files/2016/10/Sample-Allocation-in-Multivariate-Surveys.pdf.

Brito, J. A., do Nascimento Silva, P. L., Semaan, G. S. and Maculan, N., (2015a). Integer programming formulations applied to optimal allocation in stratified sampling. *Surv. Methodol.*, 41(2), pp. 427–442. Available from: https://www150.statcan.gc.ca/n1/en/pub/12-001-x/2015002/article/14249-eng.pdf?st=P7ZqwcD1.

Brito, J. A., do Nascimento Silva, P. L., Maculan, N. and Semaan, G. S., (2015b). *MultAlloc: Optimal Allocation in Stratified Sampling*. R package version 1.2.

Brito, J. A., Fadel, A. and Semaan, G. S., (2022). A genetic algorithm applied to optimal allocation in stratified sampling. *Commun. Stat. Simul. Comput.*, 51(7), pp. 3714–3732. Available from: https://doi.org/10.1080/03610918.2020.1722832.

Brito, J. A., Semaan, G. S. and Fadel, A., (2023). *BRKGA: Biased Random Key Genetic Algorithm for Optimization Problems*. R package version 0.1.0.

Chatterjee, S., (1967). A note on optimum allocation. *Scand. Actuar. J.*, 50, pp. 40–44. Available from: https://doi.org/10.1080/03461238.1967.10406206.

Clerc, M., (2012). *Standard particle swarm optimisation*, Preprint on HAL Open Archive. Available from: https://hal.science/hal-00764996v1.

Clerc, M. and Kennedy, J., (2002). The particle swarm – explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.*, 6(1), pp. 58–73. Available from: https://doi.org/10.1109/4235.985692.

Cochran, W. G., (1977). *Sampling techniques*, 3rd ed. New York: Wiley. Available from: https://books.google.lt/books?id=xbNn41DUrNwC.

Dayal, S., (1985). Allocation of sample using values of auxiliary characteristic. *J. Stat. Plan. Inference*, 11(3), pp. 321–328.

del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C. and Harley, R. G., (2008). Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Trans. Evol. Comput.*, 12(2). Available from:https://doi.org/10.1109/TEVC.2007.896686.

Garciá, J. A. D. and Cortez, L. U., (2006). Optimum allocation in multivariate stratified sampling: multi-objective programming. *Comunic. Del Cimat*, no I-06-07/28-03-2006. Available from: https://cimat.repositorioinstitucional.mx/jspui/bitstream/1008/656/1/I-06-07.pdf.

Ghasemi, M., Akbari, E., Rahimnejad, A., Razavi, S. E., Ghavidel, S. and Li, L., (2018). Phasor particle swarm optimization: a simple and efficient variant of pso. *Soft Comput.*, 23, pp. 9701–9718. Available from: https://doi.org/10.1007/s00500-018-3536-8.

Gonçalves, J. F., Resende, M. G. C., (2011). Biased random-key genetic algorithms for combinatorial optimization. *J. Heuristics*, 17(5), pp. 487–525. Available from: https://doi.org/10. 1007/s10732-010-9143-1.

Gupta, S., Haq, A. and Varshney, R., (2024). Problem of compromise allocation in multivariate stratified sampling using intuitionistic fuzzy programming. *Ann. Data Sci.*, 11, pp. 425–444. Available from: https://doi.org/10.1007/s40745-022-00410-y.

Haq, A., Ali, I. and Varshney, R., (2020). Compromise allocation problem in multivariate stratified sampling with flexible fuzzy goals. *J. Stat. Comput. Simul.*, 90(9), pp. 1557–1569. Available from: https://doi.org/10.1080/00949655.2020.1734808.

Horvitz, D. G. and Thompson, D. J., (1952). A generalization of sampling without replacement from a finite universe. *J. Am. Stat. Assoc.*, 47, pp. 663–685. Available from: https://doi.org/10.1080/01621459.1952.10483446.

Imran, M., Hashim, R. and Abd Khalid, N. E., (2013). An overview of particle swarm optimization variants. *Procedia Eng.* , 53, pp. 491–496. Available from: https://doi.org/10.1016/j.proeng.2013.02.063.

Jalil, S. A., Haq, A., Owad, A. A., Hashmi, N. and Adichwal, N. K., (2023). A hierarchical multi-level model for compromise allocation in multivariate stratified sample surveys with non-response problem, *Knowl.-Based Syst.*, 278. Available from: https://doi.org/10.1016/j.knosys.2023.110839.

Kadane, J. B., (2005). Optimal dynamic sample allocation among strata. *J. Off. Stat.*, 21(4), pp. 531–541. Available from: https://doi.org/10.1184/R1/6586808.v1.

Kennedy, J., Eberhart, R., (1995). Particle swarm optimization. *Proc. ICNN'95 – Int. Conf. Neural Netw.*, pp. 1942–1948. Available from: https://doi.org/10.1109/ICNN.1995.488968.

Khan, M. F., Ali, I. and Ahmad, Q. S., (2011). Chebyshev approximate solution to allocation problem in multiple objective surveys with random costs. *Am. J. Comput. Math.*, 01(04), pp. 247–251. Available from: https://doi.org/10.4236/ajcm.2011.14029.

Khan, M. G. M. and Ahsan, M. J., (2003). A note on optimum allocation in multivariate stratified sampling. *South Pac. J. Nat. Appl. Sci.*, 21(1), pp. 91–95. Available from: https://doi.org/10.1071/SP03017.

Khan, M. G. M., Ahsan, M. J. and Jahan, N., (1998). Compromise allocation in multivariate stratified sampling: An integer solution. *Nav. Res. Logist.*, 44(1). Available from: https://doi.org/10.1002/(SICI)1520-6750(199702)44:1<69::AID-NAV4>3.0.CO;2-K.

Khan, M. G. M., Maiti, T. and Ahsan, M. J., (2010). An optimal multivariate stratified sampling design using auxiliary information: an integer solution using goal programming approach. *J. Off. Stat.*, 26, pp. 695–708. Available from: https://www.semanticscholar.org/paper/An-optimal-multivariate-stratified-sampling-design-Khan-Maiti/a8cfea23255468fd838e09ef09b2f9976f984985.

Kish, L., (1976). Optima and proxima in linear sample designs. *J. R. Stat. Soc. Ser. A*, 139(1), pp. 80–95. Available from: https://doi.org/10.2307/2344384.

Kokan, A. R. and Khan, S., (1967). Optimum allocation in multivariate surveys : An analytical solution. *J. R. Stat. Soc. Ser. B (Methodol.)*, 29(1), pp. 115–125. Available from: https://doi.org/10.1111/j.2517-6161.1967.tb00679.x.

Kuhn, H. W. and Tucker, A. W., (1951). Nonlinear programming. *Proc. 2nd Berkeley Symp. Math. Stat. Prob.*, pp. 481 – 492.

Li, J., Sun, Y. and Hou, S., (2021). Particle swarm optimization algorithm with multiple phases for solving continuous optimization problems. *Discret. Dyn. Nat. Soc.*. Available from: https://doi.org/10.1155/2021/8378579.

Mahfouz, M. I., Rashwan, M. M. and Khadr, Z. A., (2023). Optimal Stochastic Allocation in Multivariate Stratified Sampling. *Math. Stat.*, 11(4), pp. 676–684. Available from: https://doi.org/10.13189/ms.2023.110409.

Neyman, J., (1934). On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection (with discussion). *J. R. Stat. Soc.*, 97, pp. 558–625. Available from: https://doi.org/10.2307/2342192.

Powell, M. J. D., (1994). A direct search optimization method that models the objective and constraint functions by linear interpolation. In Gomez, S. and Hennart J. P. (Eds.), *Advances in Optimization and Numerical Analysis*, pp. 51–67, Kluwer Academic, Dordrecht. Available from: https://doi.org/10.1007/978-94-015-8330-5.

R Core Team, (2023). *R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing*, Vienna, Austria. Available from: https://www.R-project.org/.

Raghav, Y. S., Haq, A. and Ali I., (2023). Multiobjective intuitionistic fuzzy programming under pessimistic and optimistic applications in multivariate stratified sample allocation problems. *PLoS ONE*, 18(4). Available from: https://doi.org/10.1371/journal.pone.0284784.

Reddy, K. G., Khan, M. G. M. and Khan, S., (2018). Optimum strata boundaries and sample sizes in health surveys using auxiliary variables. *PLoS ONE*, 13(4).

Rini, D. P., Shamsuddin, S. M. and Yuhaniz, S. S., (2011). Particle swarm optimization: Technique, system and challenges. *Int. J. Comput. Appl.*, 14(1). Available from: https://doi.org/10.5120/1810-2331.

Shannon, C. E., (1948). A Mathematical Theory of Communication. *Bell Syst. Tech. J.*, 27(3), pp. 379–423. Available from: https://people.math.harvard.edu/ ctm/home/text/ others/shannon/entropy/entropy.pdf.

Swain, A. K., (2013). A note on optimum allocation in stratified random sampling. *Invest. Oper.*, 34(2).

Tsallis, C. and Stariolo, D. A., (1996). Generalized simulated annealing. *Physica A*, 233(1), pp 395–406. Available from: https://doi.org/10.1016/S0378-4371(96)00271-3.

Varshney, R., Khan, M. G. M., Fatima, U. and Ahsan, M. J., (2014). Integer compromise allocation in multivariate stratified surveys. *Ann. Oper. Res.*, 226(1), pp. 659–668. Available from: https://doi.org/10.1007/s10479-014-1734-z.

Wesołowski, J., Wieczorkowski, R. and Wójciak, W., (2024). Recursive Neyman algorithm for optimum sample allocation under box constraints on sample sizes in strata. *Surv. Methodol.*, 50(2). Available from: https://www150.statcan.gc.ca/n1/pub/12-001-x/2024002/article/00003-eng.pdf.

Wolsey, L. A., (1998). *Integer Programming*, John Wiley & Sons, New York. Available from: https://books.google.lt/books/about/Integer$_P$*rogramming.html*?*id* = *x*7*RvQgAA CAAJredir*$_e$*sc* = *y*.

Wright, T., (2017). Exact optimal sample allocation: More efficient than Neyman. *Stat. Probab. Lett.*, 129, pp. 50–57. Available from: https://doi.org/10.1016/j.spl.2017.04. 026.

Wright, T., (2020). A general exact optimal sample allocation algorithm: With bounded cost and bounded sample sizes. *Stat. Probab. Lett.*, 165. Available from: https://doi.org/ 10.1016/j.spl.2020.108829.

Xiang, Y., Gubian, S., Suomela, B. and Hoeng, J., (2013). Generalized simulated annealing for global optimization: The GenSA package. *R J.*, 5(1), pp. 13–28.Available from: https://doi.org/10.32614/RJ-2013-002.

Yates, F., (1960). *Sampling Methods for Censuses and Surveys*, Charles Griffin and Co., London. Available from: https://archive.org/details/samplingmethodsf0000fran/page/ n5/mode/2up.