Catalogue no. 12-001-X
ISSN 1492-0921

Survey Methodology

Graphical finite population sampling

by Bardia Panahbehagh

Release date: June 29, 2026

Statistics Canada Statistique Canada

Canada

How to obtain more information

For information about this product or the wide range of services and data available from Statistics Canada, visit our website, www.statcan.gc.ca.

You can also contact us by

Email at infostats@statcan.gc.ca

Telephone, from Monday to Friday, 8:30 a.m. to 4:30 p.m., at the following numbers:

- Statistical Information Service 1-800-263-1136
- National telecommunications device for the hearing impaired 1-800-363-7629
- Fax line 1-514-283-9350

Standards of service to the public

Statistics Canada is committed to serving its clients in a prompt, reliable and courteous manner. To this end, the Agency has developed standards of service which its employees observe in serving its clients. To obtain a copy of these service standards, please contact Statistics Canada toll-free at 1-800-263-1136. The service standards are also published on www.statcan.gc.ca under "Contact us" > "Standards of service to the public."

Note of appreciation

Canada owes the success of its statistical system to a long-standing partnership between Statistics Canada, the citizens of Canada, its businesses, governments and other institutions. Accurate and timely statistical information could not be produced without their continued co-operation and goodwill.

Survey Methodology, June 2026
Vol. 52, No. 1, pp. 129-150
Statistics Canada, Catalogue No. 12-001-X

Graphical finite population sampling

Bardia Panahbehagh[1]

Abstract

This paper introduces an innovative and intuitive finite population sampling method that has been developed using a unique graphical framework. In this approach, first-order inclusion probabilities are represented as bars on a two-dimensional graph. By manipulating the positions of these bars, researchers can create a wide range of different sampling designs. This graphical visualization of sampling designs facilitates the exploration of alternative designs and may simplify certain aspects of the implementation compared to traditional mathematical algorithms. This novel approach holds significant promise for tackling complex challenges in sampling, such as achieving an optimal design. By applying a version of the greedy best-first search algorithm to this graphical approach, the potential for integrating intelligent algorithms into finite population sampling is demonstrated.

Key Words: Inclusion probabilities; Intelligent algorithm; Optimal design; Sampling design.

1. Introduction

Let $U = \{1,\ldots,N\}$ be a population of size N, and define

$$\mathcal{S} = \{s_1, s_2, \ldots, s_T\},$$

as the set of all possible samples or subsets of U. The aim is to select a random sample $\mathbb{S}$ from $\mathcal{S}$ with First-order Inclusion Probabilities (FIP) $\boldsymbol{\pi} = \{\pi_k, k \in U\}$, and size $n_{\mathbb{S}}$ where

$$\sum_{k \in U} \pi_k = E(n_{\mathbb{S}}).$$

Let $\mathbf{p} = \{p_t = \Pr(\mathbb{S} = s_t); t = 1,2,\ldots,T\}$ be a sampling design that assigns a probability to each possible sample such that

$$p_t > 0, \quad \sum_{t=1}^{T} p_t = 1, \quad \text{and} \quad \pi_k = \sum_{t; s_t \ni k} p_t \quad \text{for all} \quad k \in U. \tag{1.1}$$

The second-order inclusion probabilities (SIP) can be represented in matrix form as follows

$$\Pi = [\pi_{k\ell}]_{N \times N}, \quad \text{where} \quad \pi_{k\ell} = \sum_{t; s_t \ni k,\ell} p_t, \quad k, \ell \in U.$$

Also entropy of design $\mathbf{p}$ is defined as

$$H(\mathbf{p}) = -\sum_{t=1}^{T} p_t \ln(p_t).$$

Now consider $y_k, x_k, k \in U$ as the main and auxiliary variables respectively. The primary goal of sampling is typically to estimate parameters such as the population total, denoted by,

1. Bardia Panahbehagh, Faculty of Mathematical Sciences and Computer, Kharazmi University, Tehran, Iran. E-mail: panahbehagh@khu.ac.ir.

$$Y = \sum_{k \in U} y_k.$$

The estimation of population parameters, such as a total, as well as the estimation of the variance of parameter estimators is based on the design, more specifically the FIP and the SIP. Given $\pi_k > 0$ for all $k \in U$, an unbiased estimator of Y is the Narain-Horvitz-Thompson (Narain, 1951; Horvitz and Thompson, 1952) estimator (NHT),

$$\hat{Y} = \sum_{k \in \mathbb{S}} \frac{y_k}{\pi_k},$$

with

$$\text{var}(\hat{Y}) = \sum_{k \in U} \sum_{\ell \in U} \frac{y_k}{\pi_k} \frac{y_\ell}{\pi_\ell} (\pi_{k\ell} - \pi_k \pi_\ell). \tag{1.2}$$

Assuming that all the $\pi_{k\ell}$ are strictly greater than zero, an unbiased estimator of this variance is

$$\widehat{\text{var}}(\hat{Y}) = \sum_{k \in \mathbb{S}} \sum_{\ell \in \mathbb{S}} \frac{y_k}{\pi_k} \frac{y_\ell}{\pi_\ell} \frac{\pi_{k\ell} - \pi_k \pi_\ell}{\pi_{k\ell}}. \tag{1.3}$$

This conventional approach employs designs rooted in theoretical and mathematical algorithms, which, while well-established, may pose challenges in certain scenarios. In particular, imposing additional constraints or requirements within these conventional designs often leads to increased complexity. This complexity may manifest itself in difficulties in implementing constrained designs with a given set of FIP, or in greater challenges in calculating the corresponding SIP, both of which are essential for accurate estimation and assessment. For example, constraints such as requiring all SIP to be positive, enabling or preventing the joint selection of certain units, or selecting units according to a stream (e.g., time order or sequence) can substantially increase the complexity of the design or its implementation.

In finite population sampling, improving the efficiency of sampling designs frequently requires introducing such constraints, which can complicate probability calculations. As an example, consider Probability Proportional to Size (PPS) sampling. In this unequal probability design, sample units are selected proportionally to the size of an auxiliary variable, which leads to an efficient design when the auxiliary variable is correlated with the main variable of interest. The design with unequal probabilities with replacement was first introduced by Hansen and Hurwitz (1943), while Madow (1949), Narain (1951), and Horvitz and Thompson (1952) proposed without-replacement versions of PPS sampling. Selecting a PPS sample without replacement and with a fixed sample size can involve intricate procedures, particularly when additional design constraints are introduced. For this purpose, many different designs have been proposed, 50 of which are listed in Brewer and Hanif (1983) and Tillé (2006, 2020). All these designs can implement PPS sampling that satisfies FIP (leading to unbiased estimation of certain parameters using the NHT estimator), but they may result in different SIP, which in turn affects the precision of the NHT estimator.

Although various methods exist for implementing unequal probability, without-replacement, fixed-size designs, identifying a design that achieves higher precision under specific constraints remains a challenging and active area of research. For example, the cube method proposed by Deville and Tillé (2004) provides an elegant and efficient algorithm for balanced sampling, particularly when balancing on auxiliary variables is of primary importance. More recently, determinantal sampling designs, as explored by Loonis and Mary (2019) and Loonis (2023), offer a powerful and flexible framework, enabling exact control of inclusion probabilities through parameterization of Hermitian contracting matrices. These designs possess desirable correlation structures inherent in determinantal processes, which often lead to favorable variance properties in NHT estimators across multiple auxiliary variables. Accordingly, investigating the combinations of SIP that yield an optimal PPS sampling continues to complement these advanced methodologies and remains an area of considerable interest.

Building on these existing methods and their limitations, this paper introduces a new approach aimed at offering additional flexibility and intuitive design capabilities. The core concept of this new approach is to provide a flexible graphical method for implementing equal- and unequal-probability, without-replacement, fixed-size sampling designs. The proposed graphical framework allows for the preservation of given FIP while offering flexibility in adjusting the SIP. While a simplified version of the method corresponds to the systematic sampling approach of Madow (1949), the following sections demonstrate that, starting from a simple configuration (e.g., Madow's systematic sampling) and iteratively adjusting the SIP under a fixed FIP, the procedure can generate a broad (typically continuous) family of feasible designs with finely tuned joint probabilities. By choosing suitable update rules or objective functions, one can obtain fixed-size designs or steer the construction toward solutions that optimize a chosen criterion. This versatility enables smooth transitions between distinct designs within a single framework and connects finite population sampling with intelligent algorithmic searches for optimal designs (e.g., lower variance, improved spatial spread, or other application-driven criteria).

The remainder of the paper is organized as follows. Section 2 presents the new approach. Section 3 develops procedures for generating new designs by adjusting SIP under fixed FIP. Section 4 gives a fixed-size implementation, shows its baseline equivalence to Madow (1949), and demonstrates how the same SIP-adjustment mechanism generates alternative fixed-size designs. Section 5 introduces an intelligent method of searching for an optimal design. In Section 6 some simulations are conducted. Finally, Section 7 concludes the article with a summary and suggestions.

2. A graphical approach to finite population sampling

In this study, a graphical approach is proposed for sampling with predetermined FIP, as presented in Algorithm 1.

Algorithm 1 Graphical Finite-population Sampling (GFS)

1: Create a two-dimensional coordinate system, indicate the population units on the horizontal axis, and consider the vertical axis for the First-order Inclusion Probabilities (FIP) between 0 and 1,
2: **for** $k = 1, 2, \ldots, N$ **do**
3: Draw a bar of length π_k above point k at an arbitrary or random position such that the bar is completely between 0 and 1,
4: **end for**
5: Select a random point r between 0 and 1, plot it on the vertical axis, and draw a horizontal line, say a random line, from the selected point,
6: The final sample units are all units for which the random line passes through the bars of their First-order Inclusion Probabilities (FIP).

Figure 2.1 provides an example of Algorithm 1 with

$$\boldsymbol{\pi} = \{0.38, 0.30, 0.42, 0.65, 0.25, 0.10, 0.90\}, \tag{2.1}$$

each in a different color.

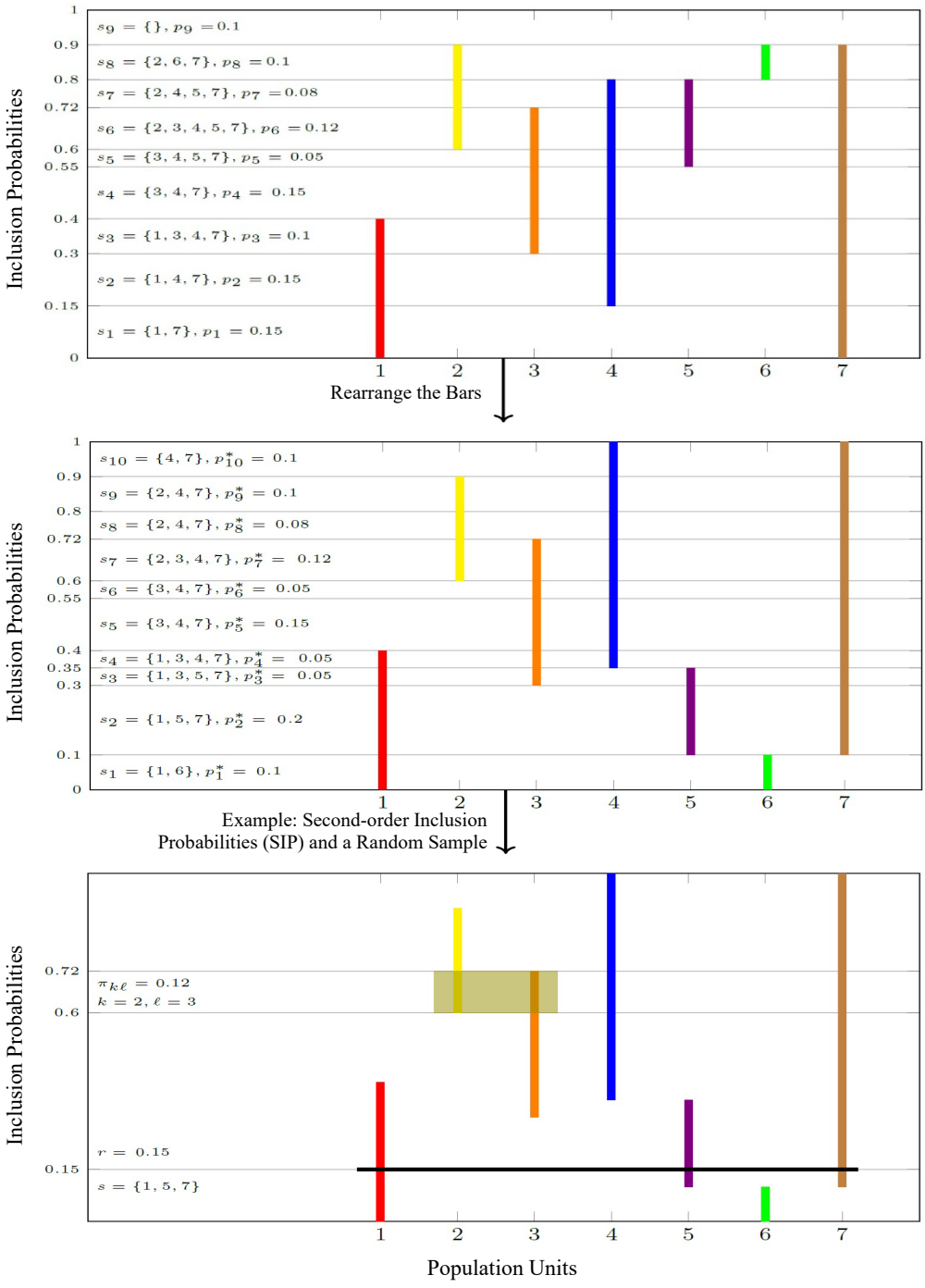
The top two plots in Figure 2.1 illustrate two different arrangements of bars, each representing a distinct design. Both arrangements adhere to the FIP constraint. The accompanying designs (samples and probabilities) to the left illustrate that any modification in the position of the bars results in a new design, as shown by the transition from design $\mathbf{p}$ with $T = 9$ in the top plot to design $\mathbf{p}^*$ with $T = 10$ in the middle plot, where the bars for π_4, π_5, π_6, and π_7 have been rearranged.

Note that when bars are rearranged, the initial configuration is treated as the reference design. For each new arrangement we do not recompute all samples from scratch; instead, we retain the previously generated sets of samples and only append the new ones corresponding to the modified bar structure. This prevents unnecessary computational cost. In some cases, however, different bar arrangements may generate identical sets of samples, in which case merging them is possible, thereby reducing the effective value of T. For instance, in the middle plot, s_5 could be merged with s_6, and s_8 with s_9, yielding $T = 8$–essentially the same design as that depicted with $T = 10$.

The bottom plot in Figure 2.1 presents two examples in design $\mathbf{p}^*$ to illustrate different concepts related to GFS:

- The first example demonstrates how to calculate the SIP for units, which can be determined by finding the intersection of the bars. In this illustration, the second-order inclusion probability for the units $k = 2$ and $\ell = 3$ is highlighted in gray, resulting in a value of $\pi_{k\ell} = 0.12$ as specified by the design.
- The second example involves drawing a random line, shown as a horizontal black line, $r = 0.15$. This random line passes through the FIP of units 1, 5 and 7, thereby selecting these three units as the final sample. This visual demonstrates how a random selection process can determine the final sample based on GFS.

Figure 2.1 The top two plots depict different arbitrary arrangements of bars with $\pi = \{0.38, 0.30, 0.42, 0.65, 0.25, 0.10, 0.90\}$ shown in different colors based on Algorithm 1, in which both arrangements respect the specified First-order Inclusion Probabilities (FIP). The designs (p and p*) are calculated and presented to the left of the plots, showing that any change in the position of the bars results in creating a new design. The bottom plot illustrates two examples: (1) The calculation of the second-order inclusion probabilities (SIP) for units $k = 2$ and $\ell = 3$, highlighted in gray, resulting in $\pi_{k\ell} = 0.12$, (2) A random line (horizontal black line) is drawn, selecting units 1, 5 and 7 as the final sample, $s = \{1, 5, 7\}$



Using GFS, one can obtain an unbiased estimator of Y while preserving the FIP. Subsequently, by rearranging the bars, the SIP can be modified, allowing control over the estimator's variance (and hence its precision).

The next Result shows that the method preserves the FIP.

Theorem 2.1. *In the GFS approach, with any arrangement of the bars,*

$$Pr(k \in \mathbb{S}) = \pi_k;\ k = 1,2,\ldots,N,\ E(n_{\mathbb{S}}) = \sum_{k \in U} \pi_k.$$

Proof. The proof follows directly from Algorithm 1.

Algorithm 1 suffers from two key limitations: a high propensity for zero SIP and variable sample size. As an example consider the top plot in Figure 2.1, where the sample size fluctuates among $n_s =$ 0, 2, 3, 4, 5, and the SIP for $k = 1,\ \ell =$ 2, 5, 6 are all zero. These deficiencies warrant further exploration, as they significantly impact the algorithm's efficiency and applicability in real-world scenarios. Subsequent sections present novel algorithms specifically developed to overcome these limitations, significantly improving the overall effectiveness of the sampling process.

3. Generating new designs by adjusting Second-order Inclusion Probabilities (SIP)

Even a zero element in the SIP matrix precludes the existence of an unbiased variance estimator for the NHT estimator. Therefore, while a design with a fully positive SIP matrix does not guarantee higher efficiency, it is still of interest to construct such designs, as they allow for unbiased variance estimation. In this section, an algorithm is proposed for the reduction of the zero elements in the SIP matrix.

To begin, please note that initially each bar represents the complete portion associated with a given π_k. A bar can be divided into smaller parts that are referred to as segments. These segments indicate the different samples to which a unit can belong (see Figure 3.1). Then, consider the y-axis partitioned into D strips, denoted by $\Delta_1, \Delta_2, \ldots, \Delta_D$, corresponding to the samples $s_1, s_2, \ldots, s_D$. Note that $D \geq T$, as some samples may appear multiple times across the y-axis partitioning. Now, select two strips, Δ_i and Δ_j, with respective heights p_i and p_j. Inside each strip, consider two smaller substrips δ_i and δ_j, both of size $v_{i,j}(\alpha) = \alpha \times \min(p_i, p_j)$ for $\alpha \in [0,\ 1]$, and define:

$$\delta_i(k) = \begin{cases} 1 & \text{If the bar for unit } k \text{ completely covers the height of substrip } \delta_i, \\ 0 & \text{otherwise.} \end{cases}$$

Next, if $s_{i-j} = s_i \setminus s_j$ is nonempty, select $k \in s_{i-j}$ and interchange the segment of π_k in δ_i with δ_j. In other words, set $\delta_i(k) = 0$ and $\delta_j(k) = 1$, which potentially leads to the following outcomes:

- sample s_i with probability p_i splitting into two samples:

  1: $s_{i,1} = s_i$, $\quad p_{i,1} = p_i - v_{i,j}(\alpha)$,

2: $s_{i,2} = s_i \setminus \{k\}, \qquad p_{i,2} = v_{i,j}(\alpha),$

- sample s_j with probability p_j splitting into two samples:

3: $s_{j,1} = s_j, \qquad p_{j,1} = p_j - v_{i,j}(\alpha),$

4: $s_{j,2} = s_j \cup \{k\}, \qquad p_{j,2} = v_{i,j}(\alpha).$

In summary, to potentially reduce zero SIP and generate new designs, in Algorithm 1 we randomly select a segment of a bar and move it to a new position, provided a vacant location is available. Details are given in Algorithm 2.

Before introducing the algorithm, it is important to clarify the use of the term "randomly" in this context. Randomly selecting two strips refers to a random mechanism that can be implemented in different ways. In the simulations presented in this paper, we use a simple random sample of size two from all possible pairs of strips. Alternatively, the selection can be made with probabilities proportional to size, where the value p_i denotes the height of strip i, which in turn represents the design weight of the respective sample. The choice of selection method may influence the rate of convergence of the algorithm toward the intended design properties.

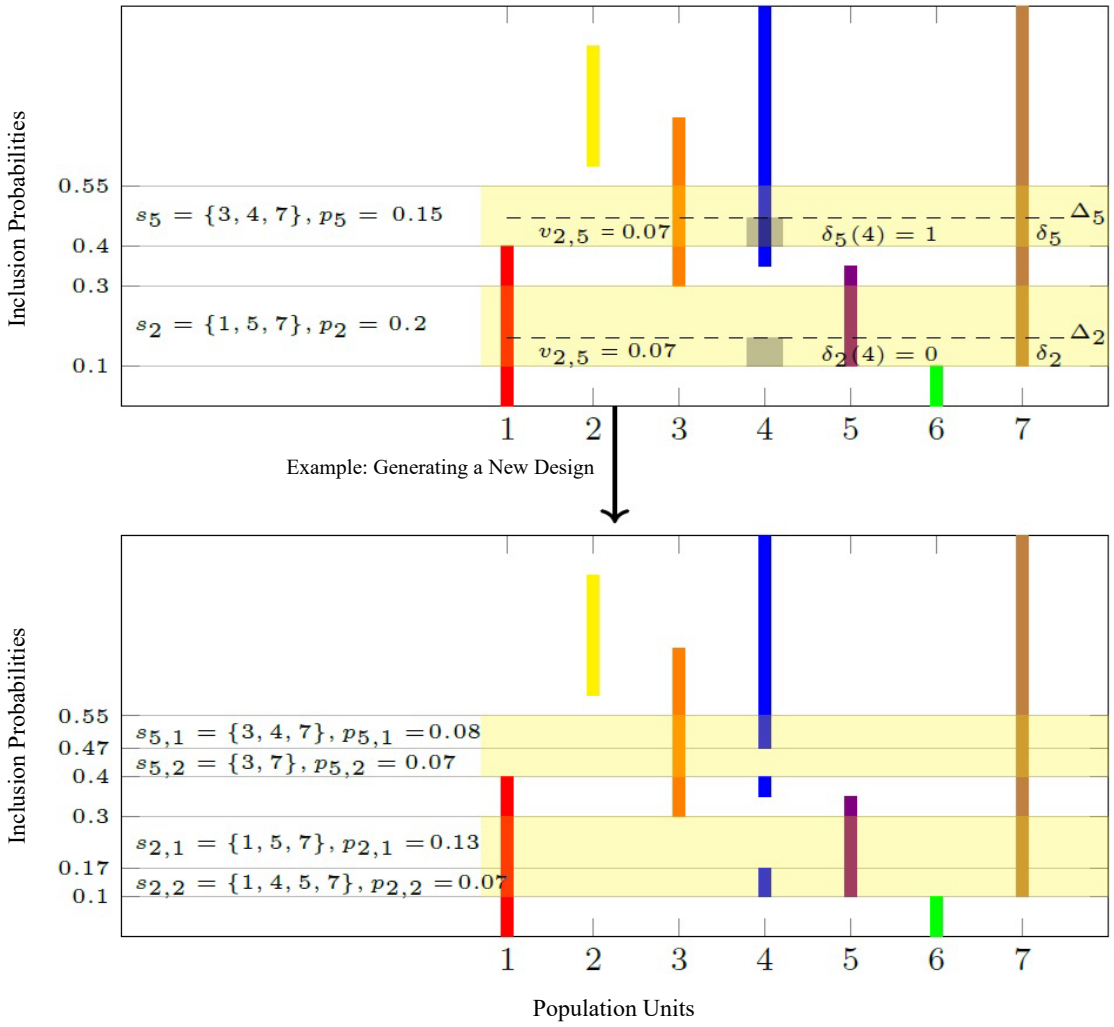
Algorithm 2 Chaotic Graphical Finite-population Sampling (GFS): Design Generator

1: Create a two-dimensional coordinate system, placing the population units on the horizontal axis and using the vertical axis for the First-order Inclusion Probabilities (FIP), ranging from 0 to 1, and build the sampling design.
2: Choose the total number of chaotic iterations *M*.
3: **for** $m = 1,2,\ldots,M$ **do**
4: Randomly select two strips Δ_i and Δ_j, and within them two substrips δ_i and δ_j of size $v_{i,j}(\alpha)$ with $0 < \alpha < 1$.
5: **if** s_{i-j} is nonempty **then**
6: Randomly select $k \in s_{i-j}$.
7: Set $\delta_i(k) = 0$ and $\delta_j(k) = 1$.
8: **end if**
9: **end for**
10: Draw a random line.
11: The final sample consists of all units for which the random line passes through their respective bars of the First-order Inclusion Probabilities (FIP).

To elucidate the core concept of Algorithm 2, examine the plots in Figure 3.1. In the top plot, consider Δ_2 and Δ_5 and within them observe that $\delta_5(4) = 1$ and $\delta_2(4) = 0$. Subsequently, we can truncate the corresponding segment of π_4 in δ_5 and relocate it to the new position in δ_2, resulting in $\delta_2(4) = 1$ and $\delta_5(4) = 0$, as depicted in the bottom plot. This implementation of Algorithm 2 with $M = 1$ generates a new design with more possible samples. For example, in the top plot, $\pi_{k\ell} = 0$ for $k = 4$ and $\ell = 5$, while in the new design, we have $\pi_{k\ell} = 0.07$ for $k = 4$ and $\ell = 5$.

It is noteworthy that, in GFS, the SIP are exactly computable. For any refinement level M, Algorithm 2 proceeds under fixed FIP until the configuration stabilizes; sampling is then carried out from this final design, and the operative SIP are those induced by that stabilized configuration.

Figure 3.1 An example of generating a new design $\mathbf{p}^*$ (the middle plot of Figure 2.1). By selecting Δ_2 and Δ_5 with heights $p_2^* = 0.2$ and $p_5^* = 0.15$, and choosing substrips δ_2 and δ_5 of size $v_{2,5}(7/15) = 0.07$, it is possible to interchange segments between substrips δ_2 and δ_5 for unit $k = 4$, which yields four samples: $s_{2,1}$, $s_{2,2}$, $s_{5,1}$, and $s_{5,2}$



In Algorithm 2, each FIP-preserving segment exchange empirically disperses joint mass over a larger set of feasible samples and tends to reduce the number of zero entries in the SIP matrix. Although monotonic improvement at every step is not claimed, this behavior suggests a progressive increase in distributional spread, motivating the following conjecture for future work on this class of algorithms.

Conjecture 3.1. *Chaotic GFS converges to Poisson sampling as* $M \to \infty$.

4. Fixed-size algorithm of Graphical Finite-population Sampling (GFS)

Building upon the resolution of the zero SIP issue in the previous section, this section addresses the challenge of random sample size within the framework of Algorithm 1.

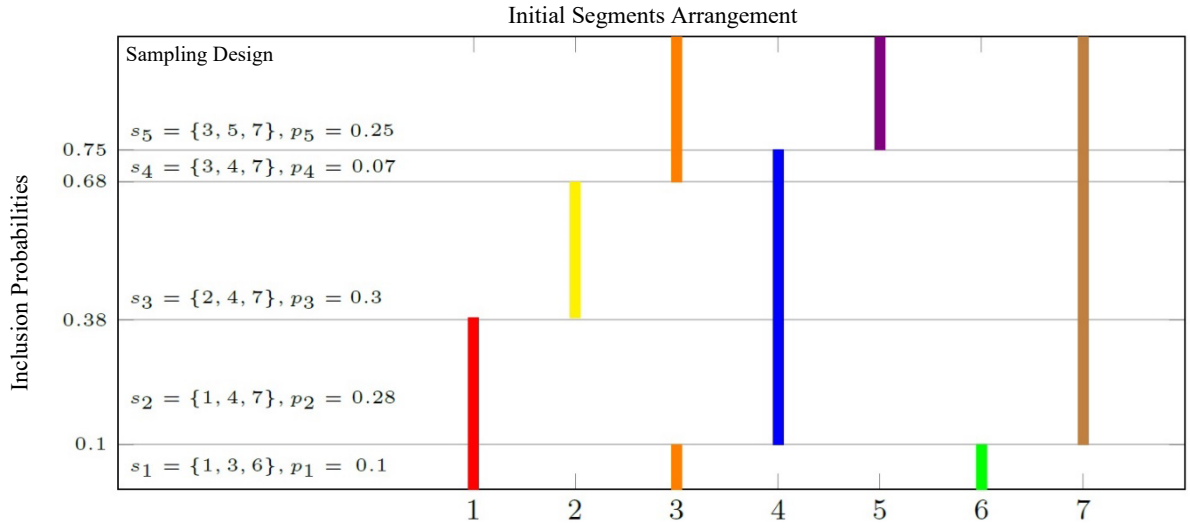
To overcome this challenge, one of the simplest approaches is to arrange the bars in a cumulative manner as follows. The first bar starts from zero to π_1, the second bar starts from π_1 to $\min(\pi_1+\pi_2,1)$, and if the minimum is 1, then the remainder of $(\pi_1+\pi_2)-1$ wraps around from zero, and so forth. Details are presented in Algorithm 3.

Algorithm 3 Fixed-size Graphical Finite-population Sampling (GFS), equivalent to Madow (1949)

1: The first bar covers $[0,\pi_1]$, and set $b_1=\pi_1$,
2: **for** $k=2,3,\ldots,N$ **do**
3: **if** $\pi_k+b_{k-1}\le 1$ **then** kth bar covers $(b_{k-1},\pi_k+b_{k-1}]$, set $b_k=\pi_k+b_{k-1}$,
4: **else** kth bar covers $(b_{k-1},1]\cup[0,\pi_k+b_{k-1}-1]$, set $b_k=\pi_k+b_{k-1}-1$,
5: **end if**
6: **end for**

Figure 4.1 presents a fixed-size version of GFS with $\boldsymbol{\pi}=\{0.38, 0.30, 0.42, 0.65, 0.25, 0.10, 0.90\}$ arranged sequentially to form a design with fixed size $n=\sum_{k=1}^{7}\pi_k=3$. In this arrangement, the FIP are aligned to maintain the fixed size constraint.

Figure 4.1 The plot depicts a fixed-size version (Algorithm 3) of Graphical Finite-population Sampling (GFS) with $\boldsymbol{\pi}=\{0.38, 0.30, 0.42, 0.65, 0.25, 0.10, 0.90\}$, arranged sequentially to create a fixed-size design



Interestingly, Algorithm 3 can be viewed as a vertical adaptation of the systematic sampling method proposed by Madow (1949). However, in the following, it will be demonstrated that unlike the systematic sampling described by Madow (1949), this novel approach possesses the capability to emulate various traditional and novel designs, including many fixed-size unequal probability sampling methods.

To generate a new design from the Madow version of GFS while preserving the FIP and the fixed-size constraint, a strategy analogous to that described in Section 3 can be employed. Specifically, two strips corresponding to two samples can be selected, and two segments of their bars can be interchanged, provided that an empty space exists in the new positions. This approach preserves fixed sample sizes while generating a new design. However, to present a complete algorithm for this process, it is necessary to define two key concepts.

Definition 4.1. Consider two substrips on the vertical axis, $\delta_i \subseteq \Delta_i$ and $\delta_j \subseteq \Delta_j$. Two segments, $\delta_i(k)$ and $\delta_j(\ell)$, are interchangeable if

$$\delta_i(k) = 1, \;\; \delta_j(\ell) = 1; \quad \delta_j(k) = 0, \;\; \delta_i(\ell) = 0.$$

Definition 4.2. Interchanging two interchangeable segments $\delta_i(k)$ and $\delta_j(\ell)$ means setting

$$\delta_i(k) = 0, \;\; \delta_j(\ell) = 0; \quad \delta_j(k) = 1, \;\; \delta_i(\ell) = 1.$$

Now, consider two strips Δ_i and Δ_j with heights p_i and p_j, and two substrips δ_i and δ_j of the same size $v_{i,j}(\alpha)$. If s_{i-j} and s_{j-i} are nonempty, select $k \in s_{i-j}$ and $\ell \in s_{j-i}$ randomly, and interchange them, potentially leading to:

- Sample s_i with probability p_i splits into two samples:
  1: $s_{i,1} = s_i, \qquad p_{i,1} = p_i - v_{i,j}(\alpha),$
  2: $s_{i,2} = s_i \cup \{\ell\} \setminus \{k\}, \quad p_{i,2} = v_{i,j}(\alpha),$
- Sample s_j with probability p_j splits into two samples:
  3: $s_{j,1} = s_j, \qquad p_{j,1} = p_j - v_{i,j}(\alpha),$
  4: $s_{j,2} = s_j \cup \{k\} \setminus \{\ell\}, \;\; p_{j,2} = v_{i,j}(\alpha).$

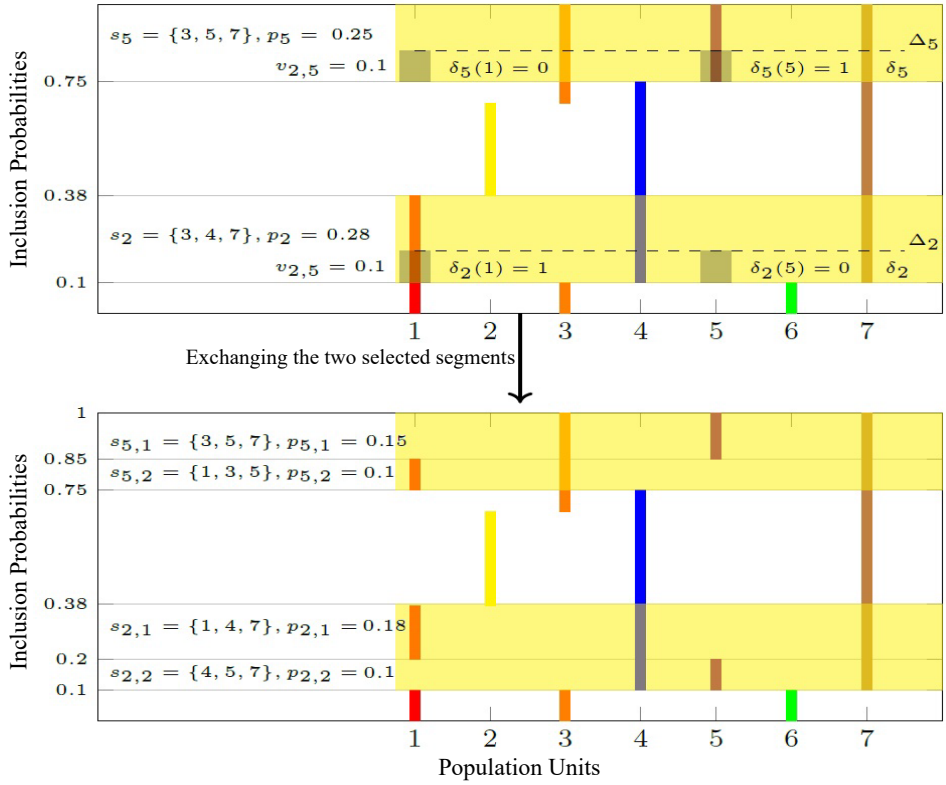
Details are presented in Algorithm 4.

Algorithm 4 Chaotic fixed-size Graphical Finite-population Sampling (GFS)

1: Implement Algorithm 3.
2: **for** $m = 1,2,\ldots,M$ **do**
3: Select randomly two strips Δ_i and Δ_j, and within them two substrips δ_i and δ_j with size $v_{i,j}(\alpha)$, $0 < \alpha < 1$,
4: **if** s_{i-j} and s_{j-i} are nonempty **then**,
5: select $k \in s_{i-j}$ and $\ell \in s_{j-i}$ randomly,
6: Interchange $\delta_i(k)$ and $\delta_j(\ell)$.
7: **end if**
8: **end for**

As an example of Algorithm 4, consider Figure 4.2, in which two strips, Δ_2 and Δ_5, are selected and by using two substrips of size $v_{2,5}(10/25) = 0.1$, segments of units 1 and 5 are interchanged to generate a new design.

Figure 4.2 **The top plot depicts a fixed-size version of Graphical Finite-population Sampling (GFS) with $\pi = \{0.38, 0.30, 0.42, 0.65, 0.25, 0.10, 0.90\}$, arranged sequentially to create a fixed-size design in which two strips, Δ_2 and Δ_5, are selected. Within these strips, two substrips of size $v_{2,5} = v_{2,5}(10/25) = 0.10$ are indicated by dashed lines. Since units 1 and 5 are interchangeable in this situation, they are interchanged, resulting in two new samples, $s_{2,2}$ and $s_{5,2}$, alongside the original samples, $s_{2,1}$ and $s_{5,1}$, depicted in the bottom plot**



The following theorem states the basic invariants of the chaotic fixed-size procedure for Algorithm 4.

Theorem 4.3. *In the chaotic fixed-size GFS approach, based on Algorithm 4,*

$$Pr(k \in \mathbb{S}) = \pi_k;\ k = 1, 2, \ldots, N,$$

$$E(n_{\mathbb{S}}) = \sum_{k \in U} \pi_k,$$

and if $E(n_{\mathbb{S}})$ is an integer, the design is fixed-size; equivalently:

$$var(n_{\mathbb{S}}) = 0.$$

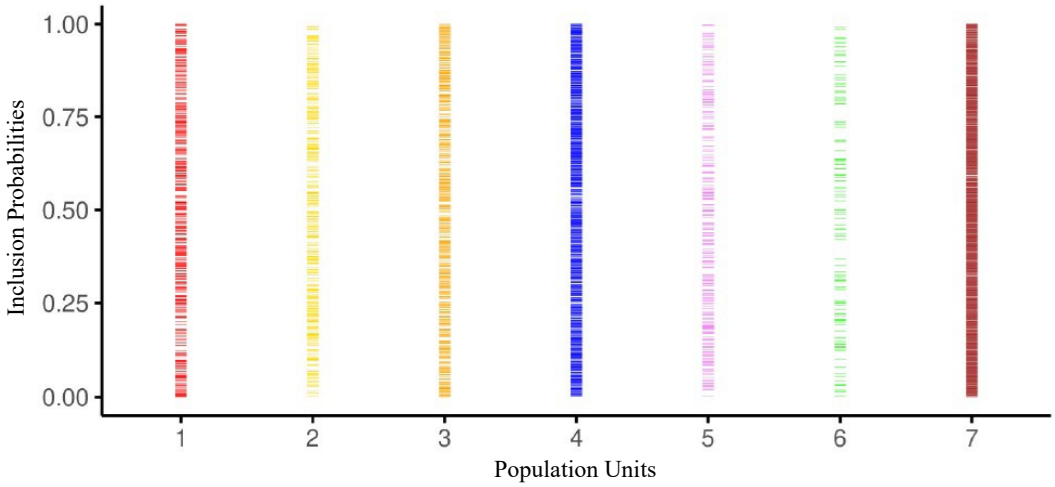
Proof. The proof follows directly from Algorithm 4.

In parallel with Conjecture 3.1, and noting that the chaotic fixed-size procedure preserves both the prescribed FIP and the sample size while empirically spreading probability mass across feasible size-n samples, it appears to increase entropy subject to the constraint $|\mathbb{S}| = n$. This motivates the following conjecture that the limiting design coincides with the conditional Poisson design–the fixed-size analogue of Poisson sampling and commonly described as maximum-entropy sampling (Tillé, 2006).

Conjecture 4.4. *Chaotic fixed-size GFS with* $M \to \infty$ *converges to maximum-entropy sampling.*

Figure 4.3 illustrates a fixed-size design corresponding to the FIP vector (2.1) generated by Algorithm 4 with $\alpha = 0.5$ and $M = 10{,}000$, characterized by broadly distributed design probabilities and substantially fewer zero SIP entries.

Figure 4.3 Implementation of Algorithm 4 on the data of population (2.1) with $\alpha = 0.5$ and $M = 10{,}000$



The framework preserves the prescribed FIP, supports fixed-size sampling, and permits the SIP to be tuned through local updates, thereby generating a broad family of feasible designs with exactly computable SIP. These properties make it natural to embed the construction within an optimization loop that seeks designs aligned with problem-specific goals while respecting the prescribed constraints.

5. Optimal Graphical Finite-population Sampling (OGFS)

As noted by Thompson and Seber (1996), there is no globally optimal sampling design for finite population sampling. The optimality of a design depends on several factors, including the structure of the main variable, the FIP, and the estimator used. In discussing optimal design, the focus is on how, given a vector of FIP, it is possible to implement an unequal probability sampling method that outperforms conventional well-known sampling designs. It is also worth noting that this discussion primarily aims to enhance the design stage when using the NHT estimator. Further research could explore improving both the design and estimation stages using GFS, which presents an interesting avenue for future investigation.

With OGFS, the GFS's ability to generate diverse designs illustrates a novel pathway, potentially offering a simple and efficient means for discovering new, effective designs.

In chaotic (random- or fixed-size) GFS, the FIP remain fixed, which facilitates the creation of designs with a very large and flexible set of possible SIP by rearranging the bars. Leveraging proper auxiliary variables, it becomes feasible to construct numerous designs. Once the position of the bars (segments) is

fixed, the design becomes fully known in GFS, enabling the calculation of criteria such as the variance, maximum error, etc., of the NHT estimator based on some auxiliary variables. Therefore, one approach is to fix a criterion, rearrange the bars to change the SIP, generate candidate designs, and select the best according to that criterion.

In the simplest case, consider two auxiliary variables x and z with some reasonable correlations with y, one for building the FIP and another for evaluating designs based on some criteria. Using

$$\pi_k = \min(cx_k, 1), \quad \text{where} \quad \sum_{k\in U}\pi_k = n, \tag{5.1}$$

leads to FIP based on x with sample size n, known as PPS. In this design, we know that $\text{var}(\hat{X}) = 0$. Then using an evaluation variable z, consider a parameter, say θ_z, rearranging the bars a large number of times based on a random or intelligent algorithm, and selecting the best arrangement to estimate θ_z based on the NHT estimator of z, results in selecting an optimal design.

Any search algorithm over designs requires a criterion to assess its proximity to the desired outcome. In the context of optimizing GFS, various design parameters can be considered as a criterion. The following equations outline three such criteria for OGFS:

$$C_1(\theta_z = Z, \mathbf{p}) = \sum_{t=1}^{T}(\hat{Z}_{s_t} - Z)^2 p_t,$$

$$C_2(\theta_z = Z, \mathbf{p}) = \sum_{t=1}^{T}|\hat{Z}_{s_t} - Z| p_t,$$

$$C_3(\theta_z = Z, \mathbf{p}) = \max_{t=1,\ldots,T}|\hat{Z}_{s_t} - Z|,$$

where

$$Z = \sum_{k=1}^{N} z_k, \quad \hat{Z}_{s_t} = \sum_{k\in s_t}\frac{z_k}{\pi_k}.$$

It is worth noting that while the process of optimizing GFS using an auxiliary variable z as a balancing target shares conceptual similarities with balanced sampling methods (Deville and Tillé, 2004; Loonis, 2023), OGFS does not explicitly enforce balance constraints at each step. Instead, it seeks designs that minimize a specified criterion (such as variance or maximum error) through intelligent rearrangement of segments, which may lead to balanced or near-balanced samples as a byproduct rather than as a direct requirement.

An Intelligent Design Search Algorithm

A variation of a greedy best-first search approach is presented to search for an optimal design. It begins with the initialization of auxiliary variables: x for constructing FIP and z for evaluating designs. An initial design is then established based on the FIP, serving as the starting point for the optimization process.

The algorithm proceeds through several iterations, say Λ, where in each cycle, N_{node} number of new candidate designs are generated by modifying the last design based on Algorithm 4. This modification

ensures the exploration of diverse configurations. Each newly generated design undergoes evaluation based on a predefined criterion, denoted as $C(\theta_z = Z, \mathbf{p})$. At the beginning, the initial design is considered the best design. Whenever a superior design is identified, it replaces the previous best design.

Two critical sets are established during this process: the open set and the closed set. The open set maintains designs, prioritized by their costs, that are yet to be explored. In contrast, the closed set keeps track of designs that have already been evaluated, ensuring that the algorithm does not redundantly process the same design multiple times. Throughout the iterations, the algorithm tracks the most effective design. Afterwards, the set of potential designs is managed to retain only the most promising candidates, replacing less efficient designs when necessary. This management ensures a focus on the most viable options for further refinement.

After all iterations are completed, the algorithm returns the best design found, representing the optimal configuration according to the defined criterion. For further details, refer to Algorithm 5.

Algorithm 5 Optimal Graphical Finite-population Sampling (OGFS): a Greedy Best-First Search Variant

Inputs: auxiliary variable x (to construct the FIP), evaluation variable z, criterion $C(\theta_z = Z, \mathbf{p})$ to be *minimized*, iteration budget Λ, branching factor N_{node}, and capacity `max_open_set_size`.
Create the `initial_design` from the FIP using Algorithm 3.
Initialize `open_set` $\leftarrow \{\}$ (priority queue keyed by criterion) and `closed_set` $\leftarrow \{\}$ (evaluated designs).
Compute `init_criterion` $\leftarrow C(\theta_z = Z, \mathbf{p}$ (`initial_design`)).
Insert (`init_criterion`, `initial_design`) into `open_set`.
`best_design` $\leftarrow$ `initial_design`; `best_criterion` $\leftarrow$ `init_criterion`.
for `iterations` $= 1$ **to** Λ **do**
    Extract and remove (`current_criterion`, `current_design`) with minimum criterion from `open_set`.
    for $i = 1$ **to** N_{node} **do**
        Generate `new_design` $\leftarrow$ APPLY-ALGORITHM 4 (`current_design`)
        `new_criterion` $\leftarrow C(\theta_z = Z, \mathbf{p}$ (`new_design`))
        if `new_design` $\notin$ `closed_set` **then**
            if `new_criterion` $<$ `best_criterion` **then**
                `best_design` $\leftarrow$ `new_design`; `best_criterion` $\leftarrow$ `new_criterion`
            end if
            if |`open_set`| $<$ `max_open_set_size` **then**
                Insert (`new_criterion`, `new_design`) into `open_set`
            else
                Let (`worst_crit`, `worst_des`) be the element with maximum criterion in `open_set`
                if `new_criterion` $<$ `worst_crit` **then**
                    Replace (`worst_crit`, `worst_des`) by (`new_criterion`, `new_design`) in `open_set`
                end if
            end if
        end if
    end for
    Add `current_design` to `closed_set`
end for
return `best_design`

6. Simulations

To evaluate Algorithm 5 in finding optimal designs, a series of simulation studies on both synthetic and real data was conducted. The focus of the study was to compare the efficiency of OGFS relative to the following designs:

- SRS: Simple Random Sampling,
- CUB: Cube Method (Deville and Tillé, 2004),
- DSD: Determinantal Sampling Design (Loonis, 2023).

OGFS was optimized using a variance criterion:

$$C(\theta_z = Z, \mathbf{p}) = \sum_{t=1}^{T} (\hat{Z}_{s_t} - Z)^2 p_t,$$

where $\hat{Z}_{s_t}$ is the NHT estimator for Z based on sample s_t.

The efficiency of estimators of Y was measured as

$$EF_{SRS} = \frac{\text{var}(\hat{Y}_{\text{SRS}})}{\text{var}(\hat{Y}_{\text{OGFS}})}, \quad EF_{DSD} = \frac{\text{var}(\hat{Y}_{\text{DSD}})}{\text{var}(\hat{Y}_{\text{OGFS}})}, \quad EF_{CUB} = \frac{\text{var}_{MC}(\hat{Y}_{\text{CUB}})}{\text{var}(\hat{Y}_{\text{OGFS}})},$$

with analogous expressions used for the efficiency of estimators of Z. The estimators $\hat{Y}_{\text{SRS}}, \hat{Y}_{\text{DSD}}$ and $\hat{Y}_{\text{CUB}}$ are NHT estimators for SRS, DSD and CUB sampling methods. For SRS, DSD, and OGFS, the variance of the NHT estimator can be computed exactly from the known first- and second-order inclusion probabilities, without the need for Monte Carlo approximation. The reported values for these methods are therefore exact. For the Cube method, however, the variance was approximated via Monte Carlo simulation, as the exact computation was not straightforward.

For this simulation, a specialized Python package was utilized, `graphical-sampling` (Panahbehagh, Mohebbi, HosseiniNasab and Moghadam, 2025), which implements the Graphical Sampling Approach. This package served as the core framework for the simulations in this section, enabling efficient and accurate implementation of the sampling techniques discussed.

All designs in this simulation section–except for SRS–use the same auxiliary variable both for defining unequal inclusion probabilities and for balancing (or optimizing) the sampling design. In the case of the Cube method, the FIP vector is additionally incorporated as a balancing variable in order to enforce fixed sample size.

Also, in all simulations, the parameters of Algorithm 5 were set to $M \sim \text{DU}(1,3)$ and $\alpha \sim \text{CU}(0.7, 0.9)$, where DU and CU denote discrete uniform and continuous uniform distributions, respectively.

These parameter ranges reflect the intended behavior of the intelligent algorithm. The parameter M controls how many local modifications are applied at each iteration. Large values make the search unstable, while $M = 1$ leads to very slow exploration. Randomizing M over $\text{DU}(1, 3)$ provides a practical balance between exploration and stability. The parameter α governs the strength of each local perturbation in

Algorithm 5. Small values produce negligible progress, whereas values close to one result in overly aggressive updates. Choosing α in $\mathrm{CU}(0.7, 0.9)$ yields sufficiently strong yet stable steps, and these settings performed robustly across all preliminary experiments. In practice, M should regulate how many changes are applied per iteration, while α should regulate their intensity.

6.1 Simulated population

For the simulated population, data were generated as follows:

- Population size: $N = 100$; sample size: $n = 10$.
- The main variable y was generated as $y \sim \mathcal{N}(100, 10^2)$.
- Auxiliary variables were generated as

$$z = b_1 y + \epsilon_z, \quad x = b_2 y + \epsilon_x, \quad b_1, b_2 \in \mathbb{R}, \quad \epsilon_z \sim \mathcal{N}(0, \sigma_z^2), \quad \epsilon_x \sim \mathcal{N}(0, \sigma_x^2),$$

  where σ_z and σ_x were selected to yield approximate target correlations of

$$\rho_{z,y}, \rho_{x,y} \in \{0.75, 0.85, 0.95\}.$$

- For x, a constant-valued vector was also considered in one scenario to implement equal-probability sampling when evaluating OGFS.

For each combination of $\rho_{z,y}$ and $\rho_{x,y}$, Algorithm 5 was run with Λ = 1,000 iterations and N_{node} = 30. The results are shown in Figure 6.1. The efficiency of OGFS relative to DSD and CUB is shown as bar charts, while its efficiency relative to SRS is indicated as values annotated above the bars. This presentation was chosen because the high efficiency values relative to SRS would have resulted in disproportionately tall bars, making it difficult to visualize the comparative performance of DSD and CUB.

Efficiency was evaluated across different combinations of correlations between the auxiliary variables x and z and the main variable y, where $\rho_{x,y}$ = 0.00 corresponds to the case of equal probability sampling. Although the main purpose of sampling is the estimation of Y, in this study the estimation of Z –which serves as the auxiliary variable used to optimize the design in OGFS and to impose balance constraints in DSD and CUB–is also computed and plotted. This allows us to evaluate the performance of each design with respect to the variable that provides the information for optimization and balancing. Then, in the analysis, the main focus is on estimating Y.

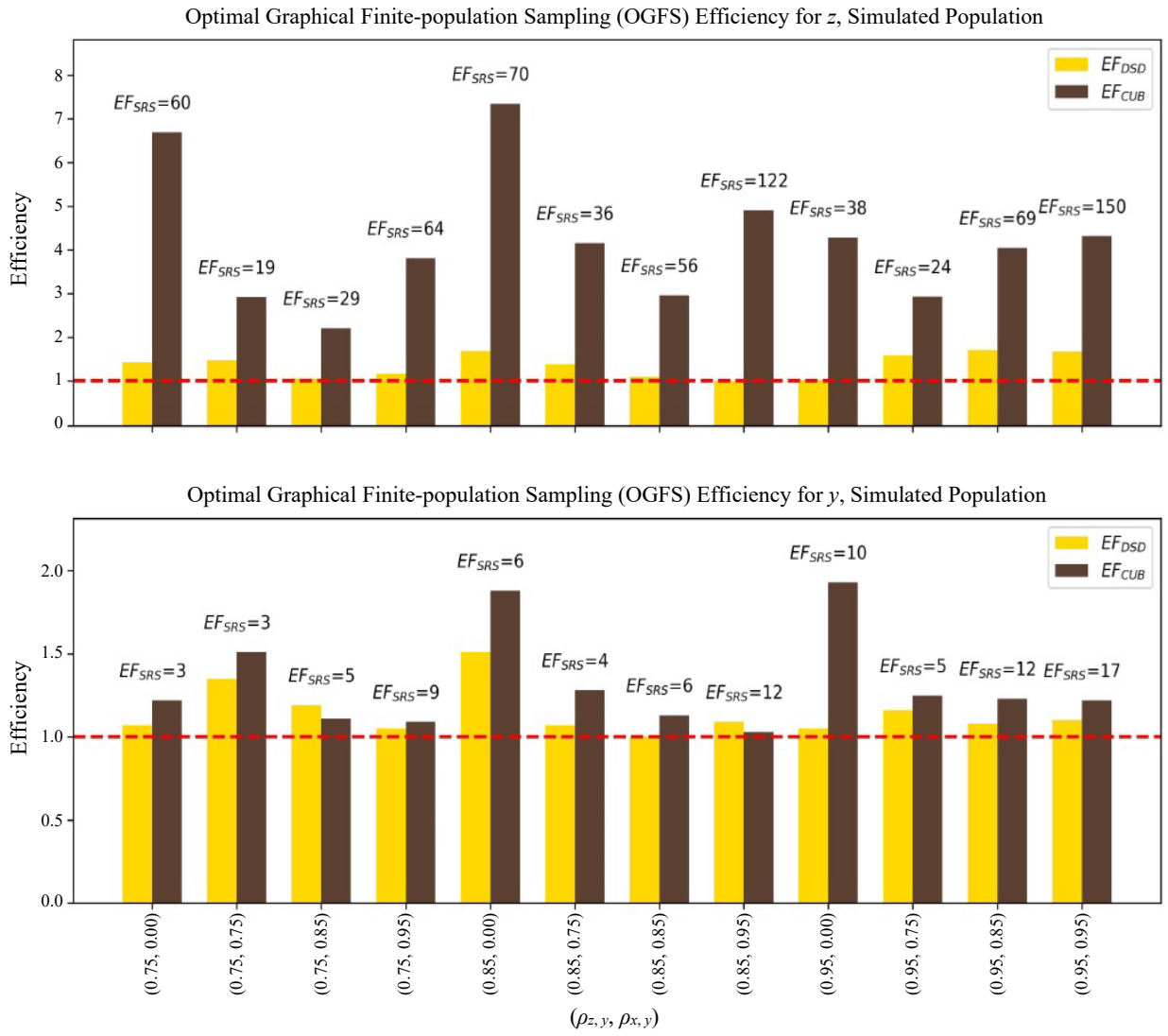
In general, OGFS demonstrated good performance when $\rho_{x,y} = 0$, representing the equal probability sampling case. In these scenarios, the correlation between z and y plays a crucial role, as the efficiency gain in estimating Y depends on how effectively information from z can support the estimation of Y. When moving to unequal probability sampling, the situation changes: the variation of the NHT estimator becomes primarily influenced by linear combinations of z_k / π_k for $k \in \mathbb{S}$, making the correlation between z_k / π_k and y_k / π_k particularly important. As demonstrated, Algorithm 5 was able to address these challenges effectively, with OGFS consistently outperforming or at least matching the performance of its

competitors, DSD and CUB. The efficiency of OGFS relative to SRS was substantial across all scenarios, and notably, DSD outperformed CUB in almost all cases.

Considering the magnitude of efficiency gains, OGFS exhibited superior performance in estimating the parameter of the auxiliary variable used for optimization, z (top plot). Furthermore, in terms of the number of scenarios where OGFS strictly outperformed its competitors, DSD and CUB, its performance was particularly noteworthy for estimating the parameter of the main variable, y (bottom plot).

Overall, these simulation studies confirm the robustness and efficiency of OGFS across all tested settings, involving various combinations of $\rho_{z,y}$ and $\rho_{x,y}$.

Figure 6.1 Efficiency of Optimal Graphical Finite-population Sampling (OGFS) relative to Simple Random Sampling (SRS), Determinantal Sampling Design (DSD), and Cube Method (CUB) across different correlation settings in simulated populations. The bars represent efficiency relative to DSD and CUB, while efficiency relative to SRS is annotated as values above the bars to enhance readability, given the large values observed for SRS. The red horizontal line indicates the baseline where efficiency equals 1. Top: efficiency for z; Bottom: efficiency for y



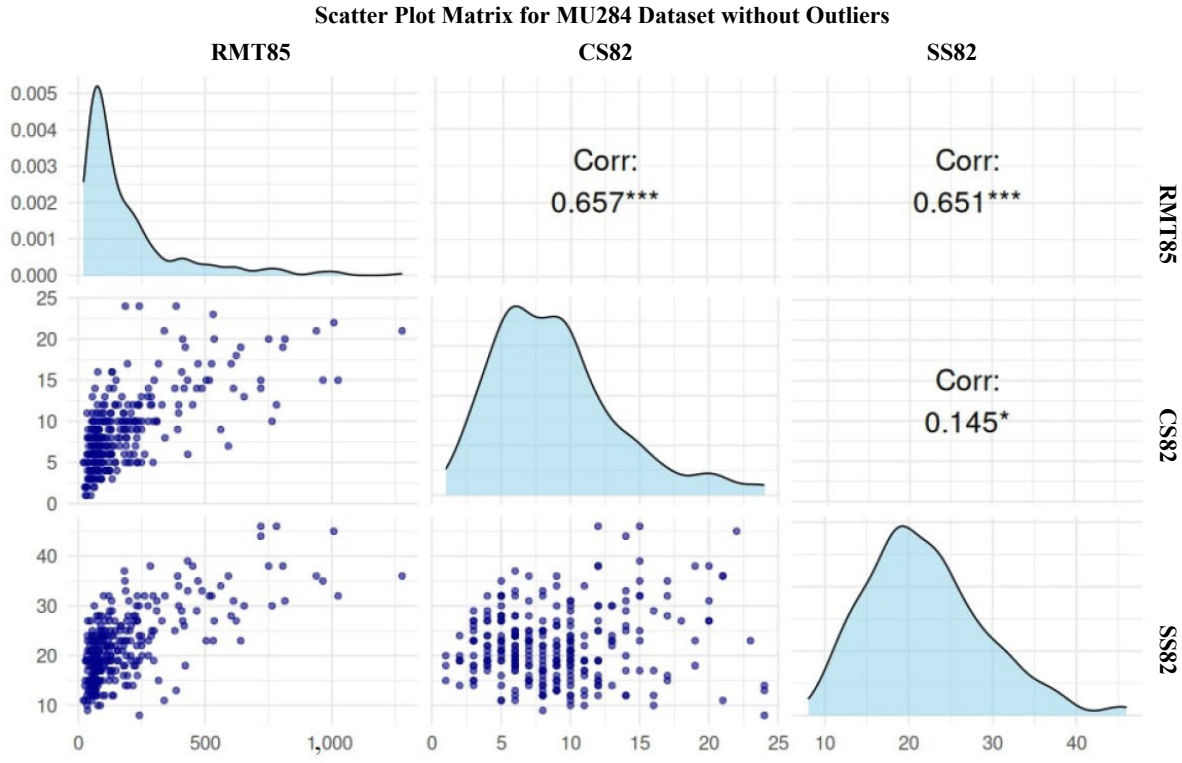
6.2 MU284 data

Now that Algorithm 5 has been evaluated across different simulated scenarios involving correlations between the main and auxiliary variables, this subsection assesses its performance in a real case study, the `MU284` dataset from Särndal, Swensson and Wretman (1992), which contains data from 284 Swedish municipalities. To improve the clarity of our visualizations and ensure the results are generalizable, three outliers were removed to prevent the findings from being driven by the influence of extreme values. This assessment considered the following variables:

- $y =$ `RMT85`: revenues from 1985 municipal taxation (in millions of kronor),
- $z =$ `CS82`: number of Conservative seats in the municipal council,
- $x =$ `SS82`: number of Social-Democratic seats in the municipal council.

Figure 6.2 presents the distributions and correlations of the selected variables. The main and auxiliary variables were chosen to ensure that the estimation of y based on x and z was meaningful given the temporal availability of the data. Both x and z exhibited moderate correlations with y ($\rho_{z,y} = 0.66$, $\rho_{x,y} = 0.65$), providing a realistic context to evaluate the effectiveness of Algorithm 5.

Figure 6.2 Scatter plot matrix and distributions for the three selected variables from the MU284 dataset after outlier removal



Note: The figure displays Pearson correlation coefficients, with significance levels indicated by symbols: * for p-value < 0.05 and *** for p-value < 0.001.

In addition to examining the effect of moderate correlation in a realistic setting, the role of N_{node}, an important parameter of Algorithm 5, was investigated. In general, aside from the inherent randomness in the search path, larger values of N_{node} increase the breadth of the search per iteration and are expected to lead to greater efficiency of OGFS. To this end, combinations of three sample sizes (n = 5, 10, 15) and three levels of N_{node} (10, 30, and 50) were considered. The number of iterations was fixed at Λ = 2,500.

The results are presented in Figure 6.3. As before, efficiencies relative to DSD and CUB are shown as bar charts, while efficiencies relative to SRS are annotated as values above the bars. The results indicate that, as expected, increasing N_{node} generally improves the efficiency of OGFS. With sufficiently large values of N_{node}, OGFS outperformed SRS, DSD, and CUB in almost all cases, despite the moderate correlations between x and y and between z and y. Here, as in the simulated population, considering the cases in which OGFS strictly outperformed its competitors, the more realistic scenario of estimating Y demonstrated stronger performance compared to the case of estimating Z, particularly when N_{node} was sufficiently large. However, in terms of the magnitude of the efficiency gains, OGFS exhibited better performance in estimating Z. This suggests that OGFS is able to leverage information from z more effectively during optimization, thereby improving both the estimation of z and y more efficiently than its rivals, DSD and CUB.

The sample size n showed a generally positive effect on the efficiency of OGFS in estimating Z, while for the main variable y, this effect was not observed as consistently. Nevertheless, across almost all sample sizes, OGFS outperformed the other designs overall.

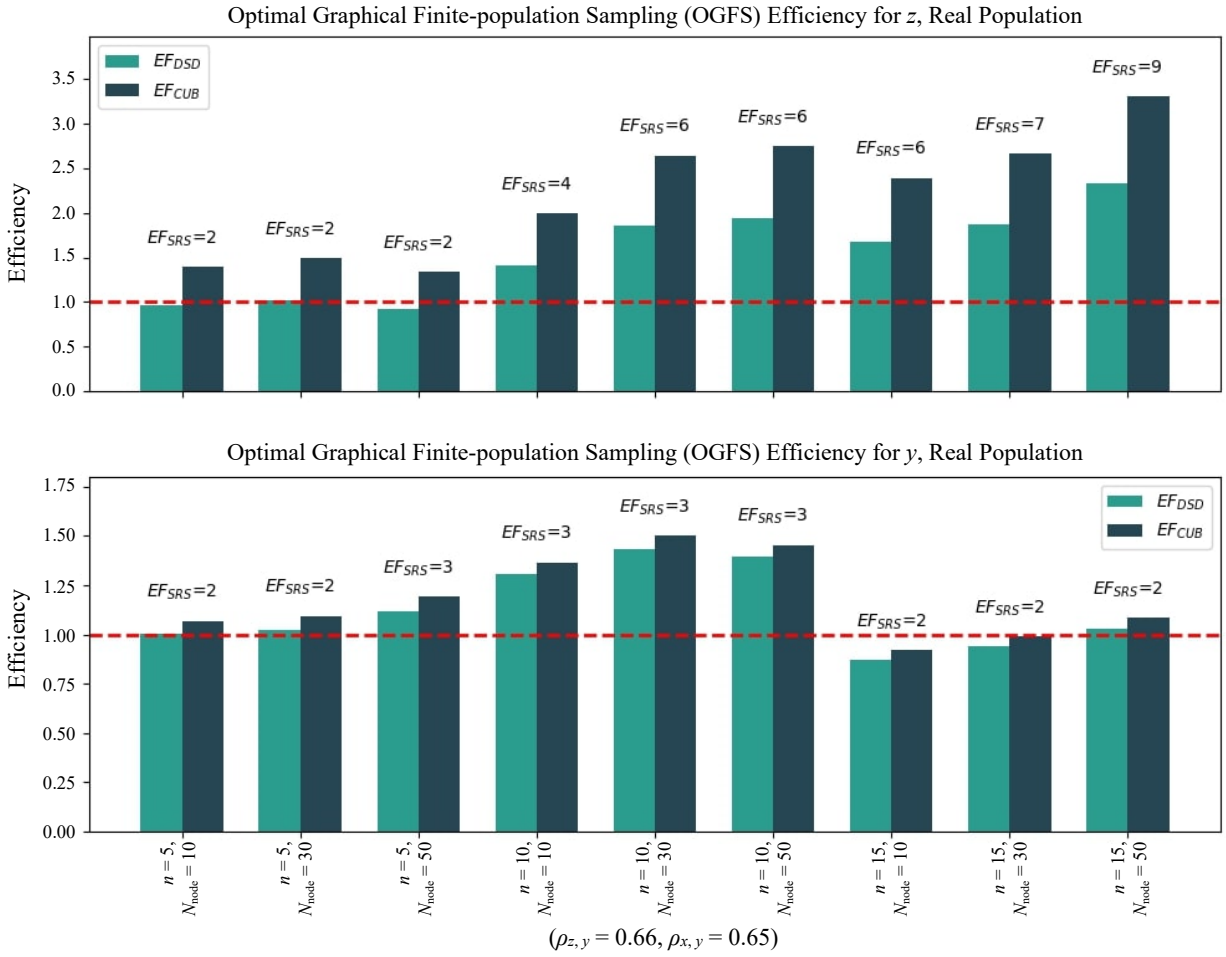
An interesting observation is that in many cases, when the algorithm was stopped, the efficiencies of OGFS were still increasing. This suggests that with a higher-performance computing configuration and sufficient compute time, it would be possible to identify even more efficient OGFS designs.

While there are many ways to further improve Algorithm 5, my aim here is not to provide an exhaustive treatment. Some potential enhancements include:

- **Multi-start strategies:** initiating the search from multiple diverse initial configurations to reduce sensitivity to starting conditions and improve exploration of the design space.
- **Adaptive open set pruning:** dynamically pruning the open set to focus on the most promising candidate designs while maintaining diversity.
- **Shocks/random injection:** introducing new initial designs mid-optimization if the process appears to be trapped in a local optimum, helping to escape suboptimal regions.
- **Dynamic adjustment of N_{node}:** varying the number of candidate designs generated at each iteration based on convergence behavior or search progress.
- **Hybrid local-global search:** combining greedy search steps with stochastic exploration methods (e.g., random swaps or perturbations) to enhance search robustness.

A comprehensive investigation of these strategies is left for future work. The focus of the present study is to illustrate that applications of innovative GFS methods, and their integration with intelligent algorithms, already demonstrate diverse advantages and notable efficiencies.

Figure 6.3 Efficiency of Optimal Graphical Finite-population Sampling (OGFS) relative to Simple Random Sampling (SRS), Determinantal Sampling Design (DSD), and Cube Method (CUB) for the MU284 dataset across different values of N_{node} and sample sizes. The bars represent efficiency relative to DSD and CUB, while efficiency relative to SRS is annotated as values above the bars to enhance readability, given the large values observed for SRS. The red horizontal line indicates the baseline where efficiency equals 1. Top: efficiency for z; Bottom: efficiency for y



7. Summary and suggestions

The GFS introduced in this study uses a graphical approach that represents a novel gateway to finite population sampling, offering a fresh perspective that reshapes conventional thinking in sampling methodology. This innovative approach fosters creativity in developing new and efficient designs by enabling researchers to explore diverse sampling strategies. Through simple segment rearrangements, GFS can generate designs ranging from the baseline systematic construction of Madow (1949)–which often induces many zero SIP entries–to alternatives with broadly distributed design probabilities and a markedly denser SIP structure. Such breadth of exploration was previously unattainable using traditional mathematical algorithms.

Two notable components of this framework, Chaotic GFS and OGFS, illustrate how GFS can either generate new designs or identify optimal configurations based on specific criteria. Additionally, the inherent flexibility of GFS naturally invites integration with a wide variety of intelligent algorithms. Beyond the greedy best-first search employed here, other optimization strategies–including genetic algorithms, simulated annealing, and particle swarm optimization–could be adapted to further improve efficiency and precision.

Despite these strengths, a notable challenge emerges with the GFS approach as population size (N) grows large. Specifically, as the total number of possible samples $(T = 2^N)$ increases exponentially, managing and preserving the full set of designs becomes computationally demanding. Nevertheless, given recent advancements in high-performance computing and the ongoing development of quantum computing technologies, these limitations are becoming increasingly manageable for small- to medium-sized populations. For larger populations and high-dimensional scenarios, further research into more sophisticated intelligent algorithms or compressed representation techniques could substantially enhance the scalability and practical applicability of GFS.

Looking ahead, a natural direction is to formalize how other standard designs (simple random, stratified, systematic, multi-stage) arise within this bar-based construction by specifying the appropriate constraints and update rules. A second priority is to investigate the conjectures formulated earlier on the limiting behavior of the chaotic updates (with and without a fixed-size constraint)–clarifying assumptions, establishing rates of concentration, and assessing robustness to implementation choices.

Given its flexibility in segment arrangement, the framework provides a unified mechanism to explore and optimize a wide array of design criteria–such as anticipated variance/efficiency, spatial spread, and distributional coverage–within a single construction. In this sense, the method offers promising avenues for generating diverse, efficient, and practically effective sampling designs.

Acknowledgements

The author expresses sincere gratitude to three of his exceptionally talented students–Mehdi H. Moghadam, Mehdi Mohebbi, and AmirMohammad HosseiniNasab–for their invaluable contributions and insightful comments, particularly during the programming stage of this research. The author is also deeply grateful to Vincent Loonis (French National Institute of Statistics and Economic Studies, INSEE) for his generous guidance and valuable discussions, which significantly contributed to the simulation study. Additionally, the author warmly thanks Prof. Yves Tillé for insightful discussions during the author's visit to the University of Neuchâtel, which helped significantly in refining the ideas presented in this paper.

References

Brewer, K.R.W. and Hanif, M. (1983). *Sampling with Unequal Probabilities*. New York: Springer.

Deville, J.-C. and Tillé, Y. (2004). Efficient balanced sampling: The cube method. *Biometrika*, 91, 893-912.

Hansen, M.H. and Hurwitz, W.N. (1943). On the theory of sampling from finite populations. *Annals of Mathematical Statistics*, 14, 333-362.

Horvitz, D.G. and Thompson, D.J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47, 663-685.

Loonis, V. (2023). Constructing all determinantal sampling designs. *Survey Methodology*, 49(2), 411-441. Available at: https://www150.statcan.gc.ca/n1/en/pub/12-001-x/2023002/article/00008-eng.pdf.

Loonis, V. and Mary, X. (2019). Determinantal sampling designs. *Journal of Statistical Planning and Inference*, 199, 60-88.

Madow, W.G. (1949). On the theory of systematic sampling, II. *Annals of Mathematical Statistics*, 20, 333-354.

Narain, R.D. (1951). On sampling without replacement with varying probabilities. *Journal of the Indian Society of Agricultural Statistics*, 3, 169-174.

Panahbehagh, B., Mohebbi, M., HosseiniNasab, A. and Moghadam, M.H. (2025). Graphical-sampling. Available at: https://pypi.org/project/graphical-sampling/.

Särndal, C.-E., Swensson, B. and Wretman, J.H. (1992). *Model Assisted Survey Sampling*. New York: Springer.

Thompson, S.K. and Seber, G.A.F. (1996). *Adaptive Sampling*. New York: John Wiley & Sons, Inc.

Tillé, Y. (2006). *Sampling Algorithms*. New York: Springer.

Tillé, Y. (2020). *Sampling and Estimation from Finite Populations*. Hoboken: Wiley.