Optimum Sample Allocation Under Box Constraints. The RNABOX Algorithm.

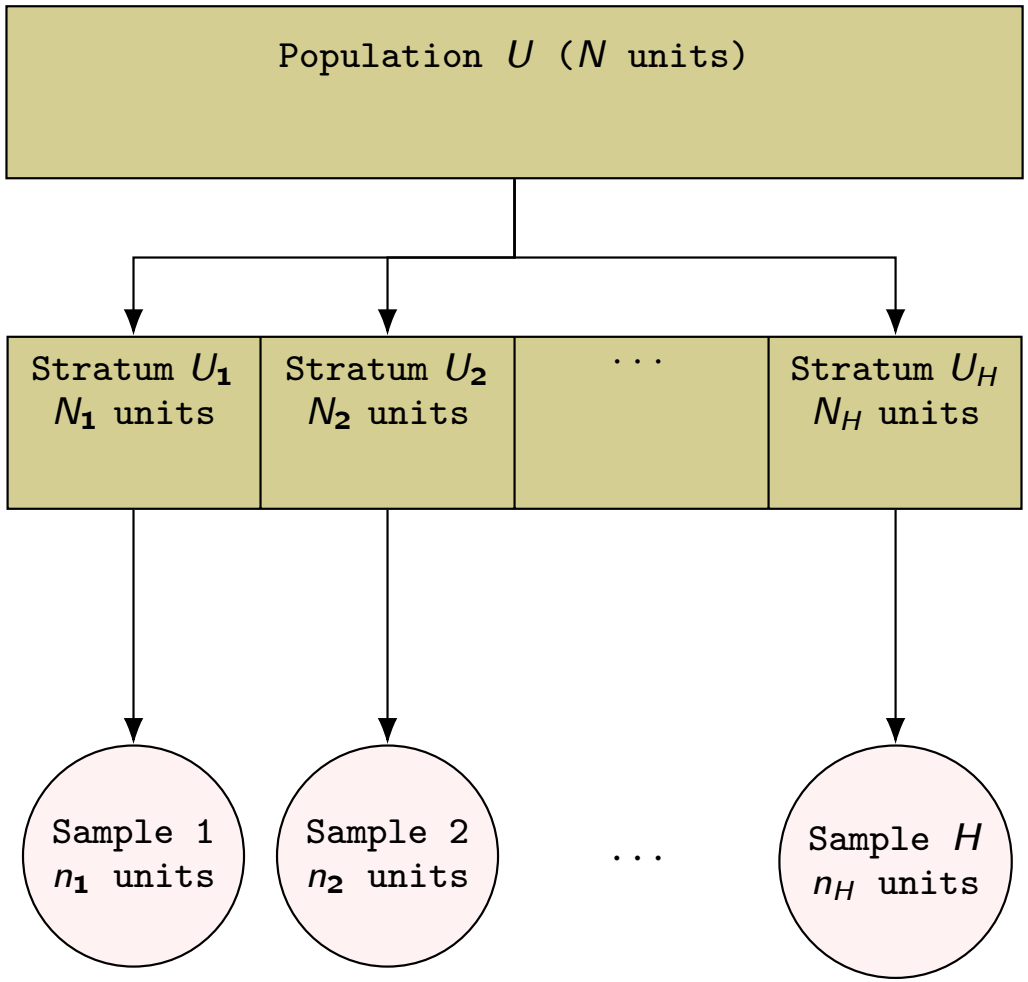
Wojciech Wójciak (with J. Wesołowski and R. Wieczorkowski)

MET2023 - Methodology of Statistical Research
July 3-5, 2023, Warsaw, Poland

Stratified sampling

- $U = \{1, 2, \ldots, N\}$ - a finite **population** consisting of N elements.
- θ - **parameter** of principal interest of a single study variable $\mathcal{Y}$ in U, e.g. $\theta = \sum_{k \in U} \mathcal{Y}(k)$.
- To **estimate** θ, we consider the **stratified sampling**.

Stratified sampling



$$U = \{1, 2, \ldots, N\}$$

$$\mathcal{H} = \{1, 2, \ldots, H\}$$
$$\sum_{h \in \mathcal{H}} N_h = N$$

$$\sum_{h \in \mathcal{H}} n_h = n$$

Classical problem of optimum sample allocation

The **stratified estimator** $\hat{\theta}_{st}$ of θ and its **variance** can be worked out for some elementary parameters (such as total or mean) and for wide range of sampling designs used within strata. Suppose that

$$Var(\hat{\theta}_{st}) = \sum_{h \in \mathcal{H}} \frac{A_h^2}{n_h} - A_0, \tag{1}$$

where A_0 and $A_h > 0$ do not depend on n_h, $h \in \mathcal{H}$ (see e.g. Särndal, Swensson and Wretman, 1992, 3.7.2-3.7.3, p. 101-106).

Problem R-BC (classical problem of optimum sample allocation)

Given set $\mathcal{H} \neq \emptyset$ and numbers $A_h > 0$, $M_h > 0$, $n > 0$, such that $M_h \leq N_h$, $h \in \mathcal{H}$, and $n \leq \sum_{h \in \mathcal{H}} M_h$,

$$\underset{\underline{x} \in \mathbb{R}_+^{|\mathcal{H}|}}{\text{minimize}} \quad f(\underline{x}) = \sum_{h \in \mathcal{H}} \frac{A_h^2}{x_h} \tag{2}$$

$$\text{subject to} \quad \sum_{h \in \mathcal{H}} x_h = n \tag{3}$$

$$x_h \leq M_h, \qquad h \in \mathcal{H}, \tag{4}$$

where $\underline{x} = (x_h,\ h \in \mathcal{H})$.

Problem of optimum sample allocation under box constraints

Problem BC (optimum non-integer allocation, finite lower and finite upper bounds)

Given set $\mathcal{H} \neq \emptyset$ and numbers $A_h > 0$, $m_h > 0$, $M_h > 0$, $n > 0$, such that $m_h < M_h \leq N_h$, $h \in \mathcal{H}$, and $\sum_{h\in\mathcal{H}} m_h \leq n \leq \sum_{h\in\mathcal{H}} M_h$,

$$\underset{\underline{x}\in\mathbb{R}_+^{|\mathcal{H}|}}{\text{minimize}} \quad f(\underline{x}) = \sum_{h\in\mathcal{H}} \frac{A_h^2}{x_h} \tag{5}$$

$$\text{subject to} \quad \sum_{h\in\mathcal{H}} x_h = n \tag{6}$$

$$m_h \leq x_h \leq M_h, \qquad h \in \mathcal{H}, \tag{7}$$

where $\underline{x} = (x_h,\ h \in \mathcal{H})$.

Selected approaches to Problem BC

- Gabler, Ganninger and Münnich (2012)
  - `noptcond()`: based on sorting, it does not guarantee (5).
- Münnich, Sachs and Wagner (2012)
  - marvellous ideas proposed, but
  - root finding methods: challenges with finding a good starting value of the parameter,
  - fixed point iteration: may not converge, not suitable when optimal solution is vertex (see below).
- Integer-valued allocation algorithms by Wright (2017), or *Capacity Scaling* by Friedrich, Münnich, de Vries and Wagner (2015)
  - great in principles since no need for rounding but relatively slow performance.

Our approach:

Sufficient optimality conditions (convex optimization)
↓
(recursive) Algorithms

Convex optimization problem and Karush-Kuhn-Tucker cond.

Problem CONV (particular type of convex optim. problem, i.e. f, h_i, g_j of *class* C^1 and g_j affine)

$$\begin{aligned} \underset{\underline{x}\in\mathbb{R}^p}{\text{minimize}} \quad & f(\underline{x}) && && (8)\\ \text{subject to} \quad & h_i(\underline{x}) = 0, && i = 1,\dots,k, && (9)\\ & g_j(\underline{x}) \le 0, && j = 1,\dots,\ell, && (10) \end{aligned}$$

$f : \mathbb{R}^p \to \mathbb{R}$, $f \in C^1$, f is convex,
$h_i : \mathbb{R}^p \to \mathbb{R}$, $h_i \in C^1$, h_i is affine, $i = 1,\dots,k$,
$g_j : \mathbb{R}^p \to \mathbb{R}$, $g_j \in C^1$, g_j is affine, $j = 1,\dots,\ell$, where $p, k, \ell \ge 1$.

Theorem (KKT necessary and sufficient conditions for CONV)

$\underline{x}^* \in \mathbb{R}^p$ solves CONV iff there exist numbers $\lambda_i \in \mathbb{R}$, $i = 1,\dots,k$, and $\mu_j \ge 0$, $j = 1,\dots,\ell$, s.t.

$$\frac{\partial}{\partial x_h} f(\underline{x}^*) + \sum_{i=1}^{k} \lambda_i \frac{\partial}{\partial x_h} h_i(\underline{x}^*) + \sum_{j=1}^{\ell} \mu_j \frac{\partial}{\partial x_h} g_j(\underline{x}^*) = 0, \quad h = 1,\dots,p,$$

$$h_i(\underline{x}^*) = 0, \quad i = 1,\dots,k,$$
$$g_j(\underline{x}^*) \le 0, \quad j = 1,\dots,\ell,$$
$$\mu_j g_j(\underline{x}^*) = 0, \quad j = 1,\dots,\ell,$$

where $p, k, \ell \ge 1$.

Set function $s(\mathcal{L},\mathcal{U})$ and $(\mathcal{L},\mathcal{U})$-allocation vector

Notation: $\mathcal{V}^c = \mathcal{H} \setminus \mathcal{V}$, for any $\mathcal{V} \subseteq \mathcal{H}$.

Definition ($s(\mathcal{L},\mathcal{U})$)

For disjoint $\mathcal{L},\mathcal{U}$, such that $\mathcal{L} \cup \mathcal{U} \subsetneq \mathcal{H}$ we define a set function s as

$$s(\mathcal{L},\mathcal{U}) = \frac{n - \sum_{h\in\mathcal{L}} m_h - \sum_{h\in\mathcal{U}} M_h}{\sum_{h\in(\mathcal{L}\cup\mathcal{U})^c} A_h} \tag{11}$$

Definition ($(\mathcal{L},\mathcal{U})$-allocation vector)

Let $\mathcal{L},\mathcal{U}$ be disjoint, such that $\mathcal{L} \cup \mathcal{U} \subseteq \mathcal{H}$. Vector $\underline{x}^{(\mathcal{L},\mathcal{U})} = (x_h^{(\mathcal{L},\mathcal{U})},\ h \in \mathcal{H})$ with entries of the form

$$x_h^{(\mathcal{L},\mathcal{U})} = \begin{cases} m_h, & h \in \mathcal{L} \\ M_h, & h \in \mathcal{U} \\ A_h\, s(\mathcal{L},\mathcal{U}) \in (m_h, M_h), & h \in (\mathcal{L}\cup\mathcal{U})^c, \end{cases} \tag{12}$$

will be called the $(\mathcal{L},\mathcal{U})$-allocation vector.

Note: $s(\mathcal{L},\mathcal{U})$ is not defined in case of $\mathcal{L} \cup \mathcal{U} = \mathcal{H}$. But then, the last line of (12) does not apply as $(\mathcal{L}\cup\mathcal{U})^c = \emptyset$.

Regular vs. vertex allocation

- Problem BC becomes trivial if:
  - $n = \sum_{h \in \mathcal{H}} m_h$, then $\underline{x}^* = (m_h,\ h \in \mathcal{H})$,
  - $n = \sum_{h \in \mathcal{H}} M_h$, then $\underline{x}^* = (M_h,\ h \in \mathcal{H})$.
- These two are boundary cases of the **vertex allocation**, i.e. an allocation $\underline{x}^* = (x_h^*,\ h \in \mathcal{H})$ of the form

$$x_h^* = \begin{cases} m_h, & h \in \mathcal{L} \\ M_h, & h \in \mathcal{U}, \end{cases}$$

  where $\mathcal{L} \cup \mathcal{U} = \mathcal{H}$ and $\mathcal{L} \cap \mathcal{U} = \emptyset$.
- It is called a vertex allocation as $\underline{x}^*$ is a vertex of the hyper-rectangle $\times_{h \in \mathcal{H}} [m_h,\ M_h]$.
- A non-vertex allocation $\underline{x}^*$ will be called a **regular allocation**.

Sufficient optimality conditions for Problem BC

It appears that the solution to Problem BC is necessarily of the form (12).

Theorem (Optimality conditions)

The optimization Problem BC has the unique solution $\underline{x}^* = \underline{x}^{(\mathcal{L},\mathcal{U})}$, *with* $\mathcal{L},\mathcal{U} \subseteq \mathcal{H}$ *such that one of the following two cases holds:*

1. $\mathcal{L} \cup \mathcal{U} \subsetneq \mathcal{H}$ *and*

$$\begin{aligned} \mathcal{L} &= \left\{ h \in \mathcal{H} : \; s(\mathcal{L},\mathcal{U}) \leq \frac{m_h}{A_h} \right\}, \\ \mathcal{U} &= \left\{ h \in \mathcal{H} : \; s(\mathcal{L},\mathcal{U}) \geq \frac{M_h}{A_h} \right\}. \end{aligned} \tag{13}$$

   Then, $\underline{x}^{(\mathcal{L},\mathcal{U})}$ *is a regular allocation.*

2. $\mathcal{L} \cup \mathcal{U} = \mathcal{H}$ *and*

$$\max_{h \in \mathcal{U}} \frac{M_h}{A_h} \leq \min_{h \in \mathcal{L}} \frac{m_h}{A_h}, \tag{14}$$

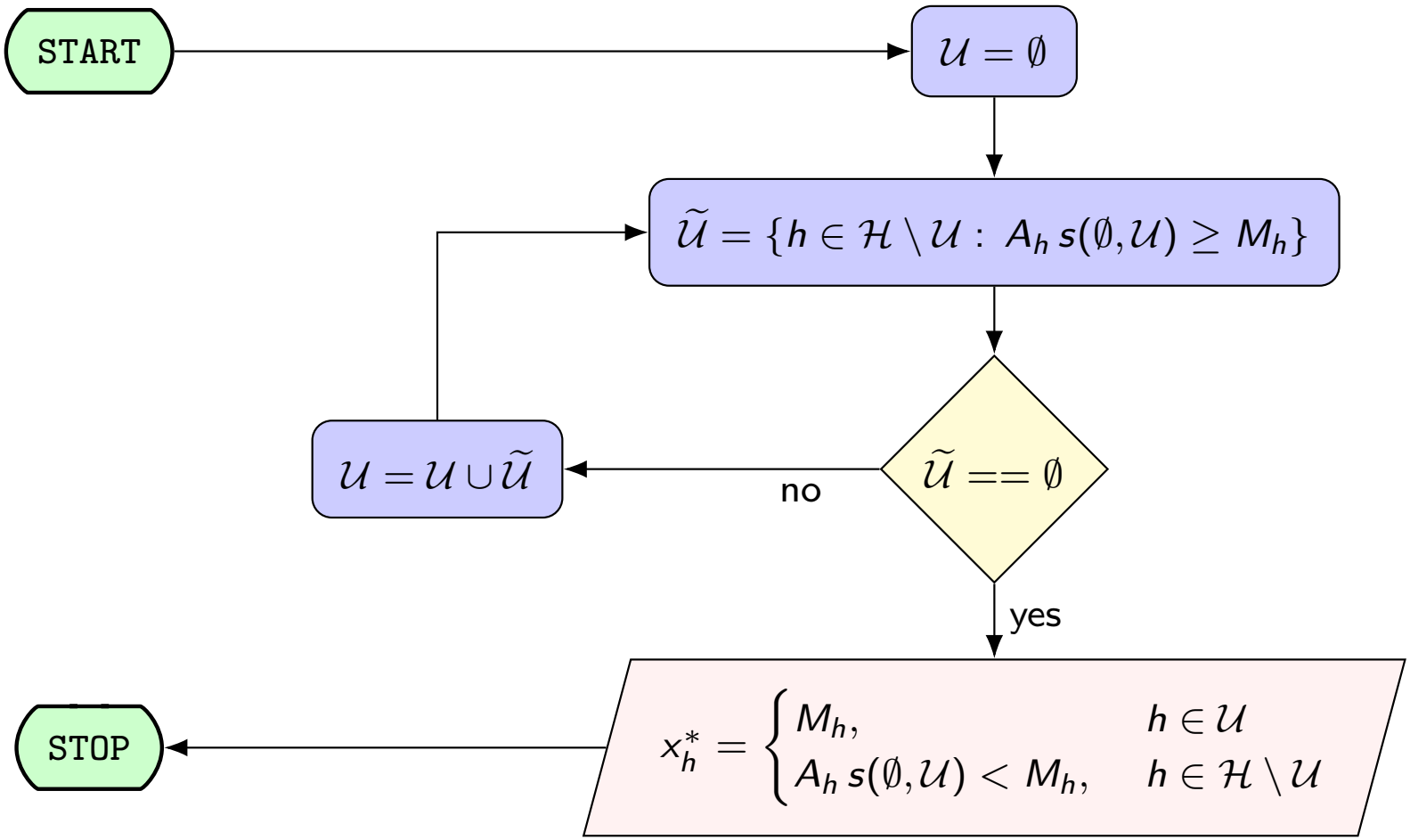
$$\sum_{h \in \mathcal{L}} m_h + \sum_{h \in \mathcal{U}} M_h = n. \tag{15}$$

   Then, $\underline{x}^{(\mathcal{L},\mathcal{U})}$ *is a vertex allocation.*

How to determine sets $\mathcal{L}$, $\mathcal{U}$?

Recursive Neyman Algorithm (RNA) for Problem R-BC

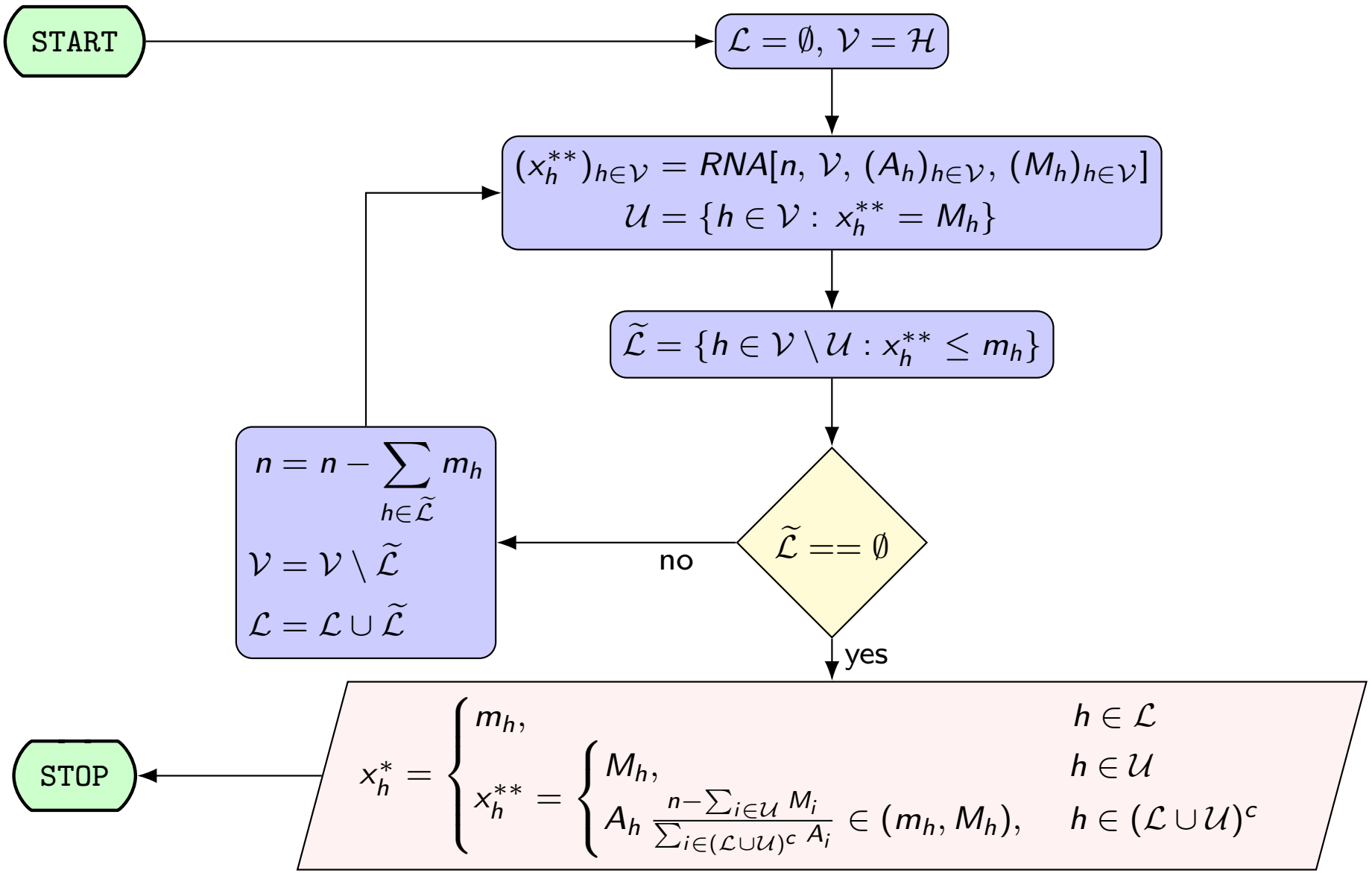
Input variables: $\mathcal{H}$, $(A_h)_{h\in\mathcal{H}}$, $(M_h)_{h\in\mathcal{H}}$, n



For more details, see e.g. Särndal et al. (1992, Remark 12.7.1, p. 466) or Wesołowski, Wieczorkowski and Wójciak (2021).

RNABOX for Problem BC

Input variables: $\mathcal{H}$, $(A_h)_{h\in\mathcal{H}}$, $(m_h)_{h\in\mathcal{H}}$, $(M_h)_{h\in\mathcal{H}}$, n



Example of RNABOX on a population with 10 strata

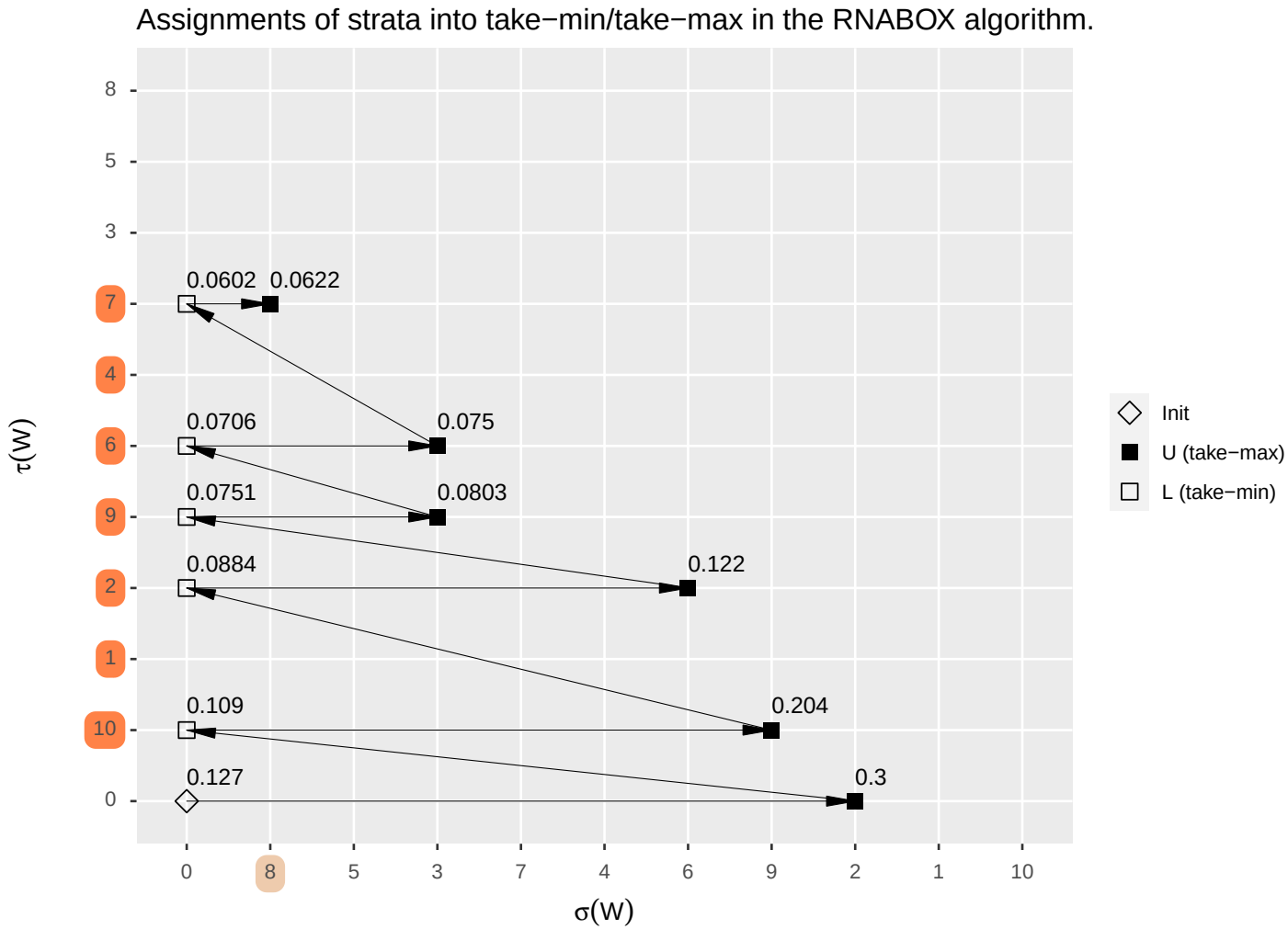
Settings:

- $\theta = \sum_{k \in U} \mathcal{Y}(k)$,
- $\hat{\theta}_{st}$ - stratified π estimator of θ,
- stratified simple random sampling without replacement design in all strata,
- $n = 5110$.

h	A_h	m_h	M_h	take-min	take-max	take-Neyman	$\underline{x}^*$
1	2700	750	900	✺			750
2	2000	450	500	✺			450
3	4200	250	300			✺	261.08
4	4400	350	400	✺			350
5	3200	150	200			✺	198.92
6	6000	550	600	✺			550
7	8400	650	700	✺			650
8	1900	50	100		✺		100
9	5400	850	900	✺			850
10	2000	950	1000	✺			950
SUM		5000	5600	7	1	2	5110

$Var(\hat{\theta}_{st}) = 33791.45$

Example of RNABOX on a population with 10 strata



Numbers above the squares are the values of set function $s(\mathcal{L},\mathcal{U})$ for $\mathcal{L}$ and $\mathcal{U}$ with elements indicated by the coordinates of a given square. For example, for ■ at coordinates $(8,7)$, we have $s(\{10,1,2,9,6,4,7\},\{8\}) = 0.0622$.

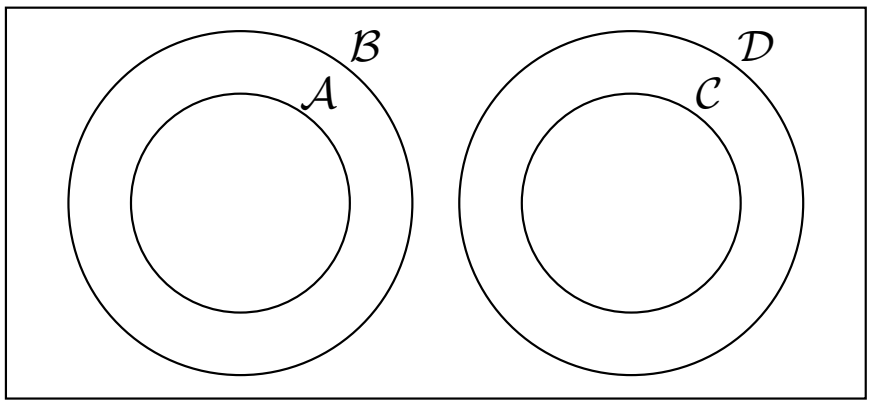
RNABOX optimality theorem

Theorem (RNABOX optimality)

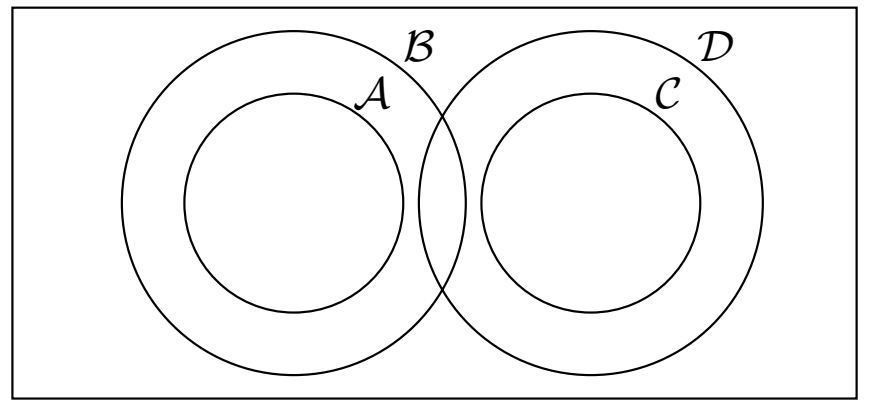
The RNABOX algorithm provides the optimal solution to optimization Problem BC.

Important and interesting properties of some interim objects

1



2



Lemma

Let set function s be as defined in (11) *and* $\mathcal{A} \subseteq \mathcal{B} \subsetneq \mathcal{H}$, $\mathcal{C} \subseteq \mathcal{D} \subsetneq \mathcal{H}$.

1. *If* $\mathcal{B} \cup \mathcal{D} \subsetneq \mathcal{H}$ *and* $\mathcal{B} \cap \mathcal{D} = \emptyset$, *then*

$$s(\mathcal{A},\mathcal{C}) \geq s(\mathcal{B},\mathcal{D}) \quad \Leftrightarrow \quad s(\mathcal{A},\mathcal{C}) \left(\sum_{h \in \mathcal{B}\setminus\mathcal{A}} A_h + \sum_{h \in \mathcal{D}\setminus\mathcal{C}} A_h \right) \leq \left(\sum_{h \in \mathcal{B}\setminus\mathcal{A}} m_h + \sum_{h \in \mathcal{D}\setminus\mathcal{C}} M_h \right).$$

2. *If* $\mathcal{A} \cup \mathcal{D} \subsetneq \mathcal{H}$, $\mathcal{A} \cap \mathcal{D} = \emptyset$, $\mathcal{B} \cup \mathcal{C} \subsetneq \mathcal{H}$, $\mathcal{B} \cap \mathcal{C} = \emptyset$, *then*

$$s(\mathcal{A},\mathcal{D}) \geq s(\mathcal{B},\mathcal{C}) \quad \Leftrightarrow \quad s(\mathcal{A},\mathcal{D}) \left(\sum_{h \in \mathcal{B}\setminus\mathcal{A}} A_h - \sum_{h \in \mathcal{D}\setminus\mathcal{C}} A_h \right) \leq \left(\sum_{h \in \mathcal{B}\setminus\mathcal{A}} m_h - \sum_{h \in \mathcal{D}\setminus\mathcal{C}} M_h \right).$$

Important and interesting properties of some interim objects

Iteration index r:

- $\mathcal{U}_r$, $\mathcal{L}_r$, $\widetilde{\mathcal{L}}_r$ denote sets $\mathcal{U}$, $\mathcal{L}$, $\widetilde{\mathcal{L}}$ respectively, in the r-th iteration of the *RNABOX* algorithm, at the moment after Step 3 ($\widetilde{\mathcal{L}}$) and before Step 4 (decision), $r = 1, \ldots, r^*$.
- $r^* \leq |\mathcal{H}| + 1 < \infty$ indicates the final iteration of the algorithm.

Lemma

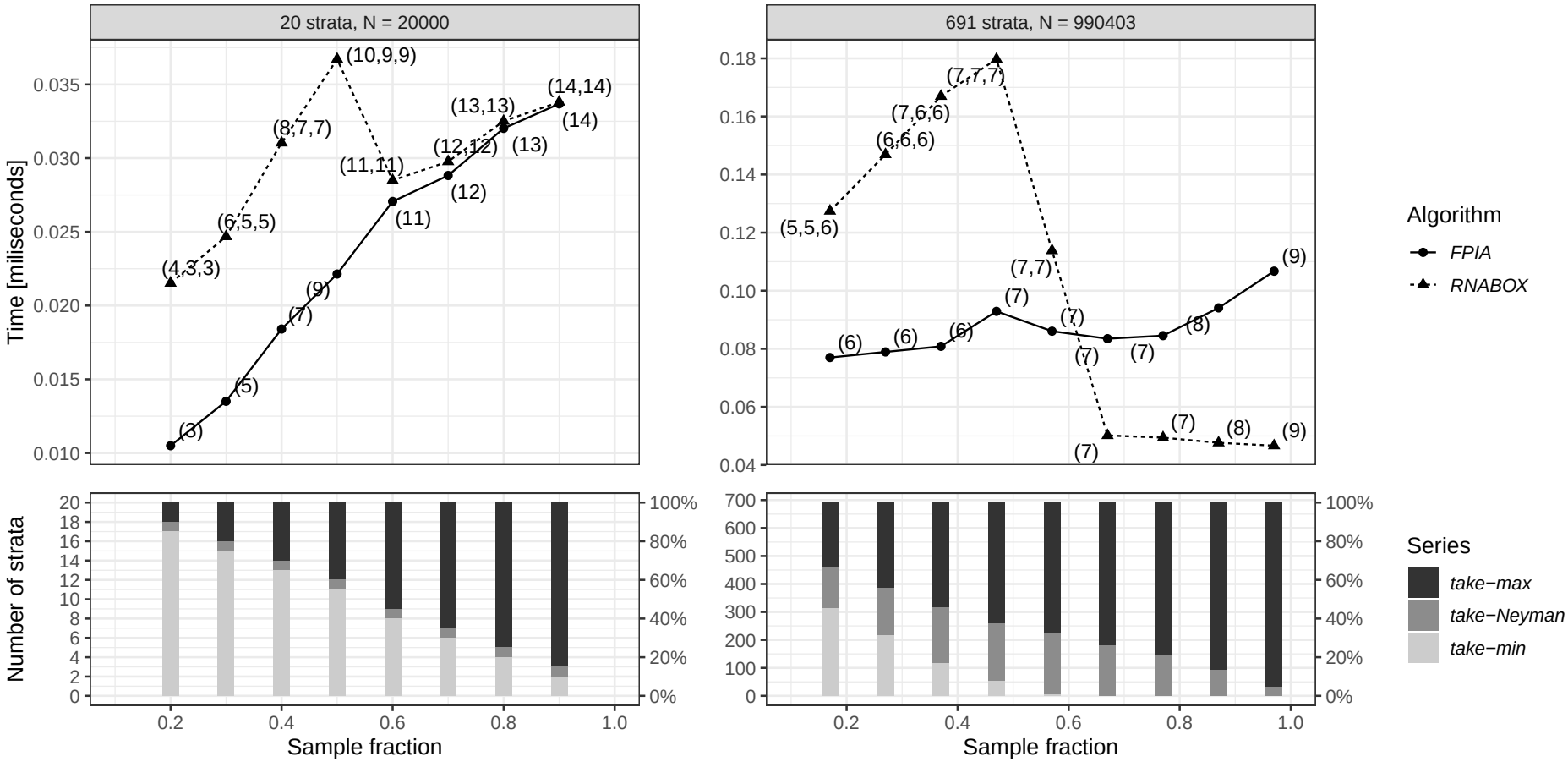
If the optimal solution to Problem BC is a **regular** *solution, then*

$$s(\mathcal{L}_{r-1}, \mathcal{U}_{r-1}) \geq s(\mathcal{L}_r, \mathcal{U}_r), \qquad r = 2, \ldots, r^* \geq 2. \tag{16}$$

Lemma

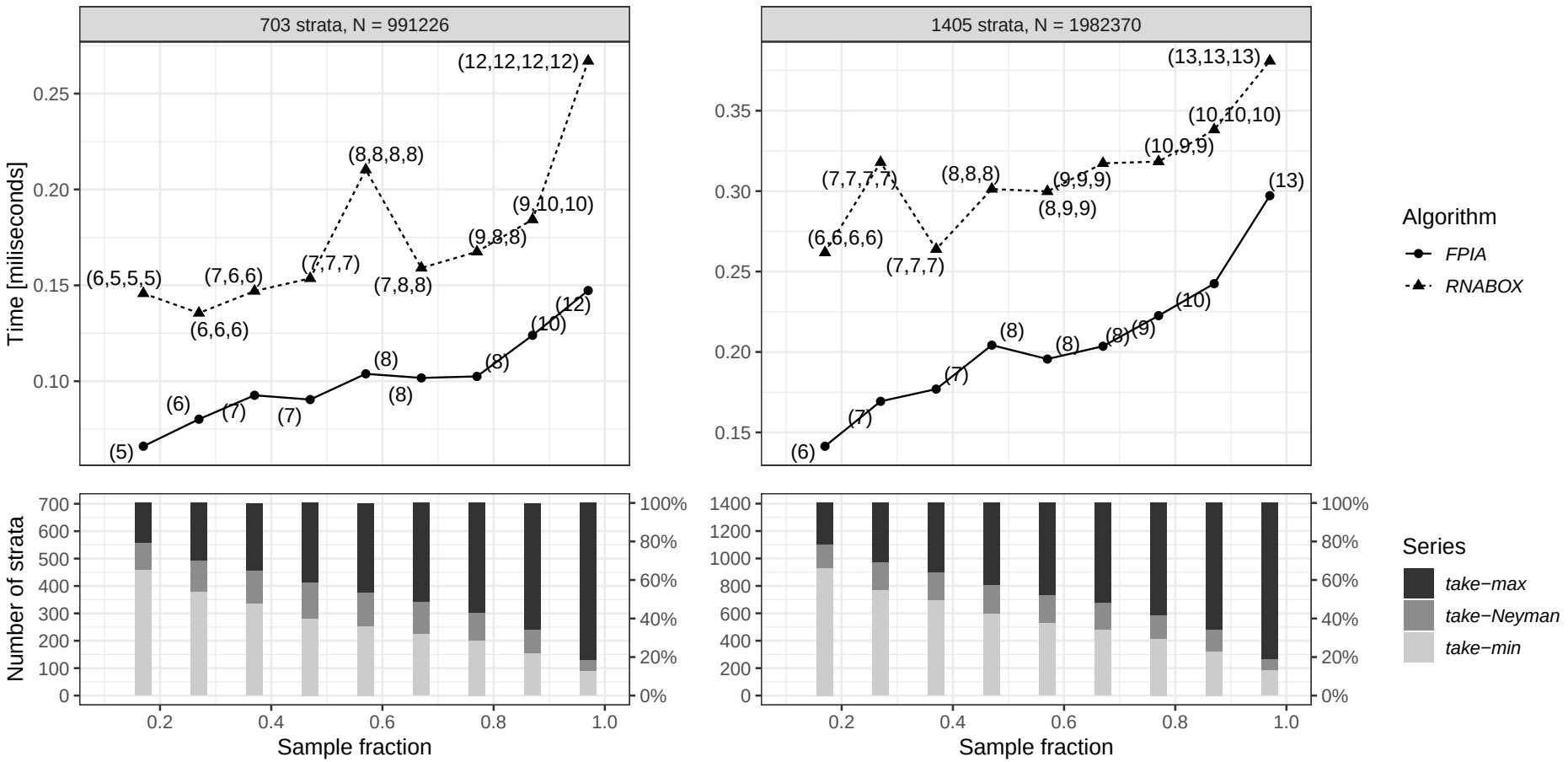
$$\mathcal{U}_{r-1} \supseteq \mathcal{U}_r, \qquad r = 2, \ldots, r^* \geq 2. \tag{17}$$

Comparison for 20 and 691 strata



FPIA - our R implementation of a fixed point iteration algorithm by Münnich, Sachs and Wagner (2012).
Number of iterations are given in brackets. For *RNABOX* we report a vector, of which i-th element gives a number of iterations of the *RNA* at the i-th iteration of the *RNABOX*.

Comparison for 703 and 1405 strata



Key features

- Provides **functions** that **solve the optimum allocation** Problem BC (and some of its relaxed versions).
- Employs best coding practices to meet specific requirements of modern computing tools:
  - **quality** (code styling, documentation, vignettes, version control) and
  - **validation** (extensive unit tests, argument assertions).
- Data structures: **simple numerical vectors**.
- Almost no **dependency** on external R packages.
- **CRAN**: https://cran.r-project.org/package=stratallo
- **GitHub**: https://github.com/wwojciech/stratallo

User functions:

- Allocation functions: `opt()`, `optcost()`.
- Helper functions: `ran_round()`, `round_oric()`, `var_st()`, `var_st_tsi()`, `asummary()`.

Example of use

opt(), var_st(), asummary().

```
> # Example population with 4 strata.
> a <- c(3000, 4000, 5000, 2000)
> # Lower bounds.
> m <- c(100, 90, 500, 50)
> # Upper bounds.
> M <- c(300, 400, 800, 90)
> # Total sample size.
> n <- 1285
>
> # Optimal allocation under box constraints.
> (xopt <- opt(n, a, m, M))
[1] 297.8571 397.1429 500.0000  90.0000
>
> # Value of the variance of the stratified pi estimator.
> var_st(xopt, a, a0 = 20)
[1] 164928
>
> # Allocation summary.
> asummary(xopt, a, m, M)
                a   m    M   allocation take_m take_M take_Neyman
Stratum_1 3000 100  300        297.9                          *
Stratum_2 4000  90  400        397.1                          *
Stratum_3 5000 500  800        500.0      *
Stratum_4 2000  50   90         90.0               *
SUM             740 1590       1285.0      1      1           2
>
> # Rounding.
> round_oric(xopt)
[1] 298 397 500  90
```

Optimum Sample Allocation Under Box Constraints. The RNABOX Algorithm.

1. Problem BC

$$\begin{aligned} \underset{\underline{x}\in\mathbb{R}_+^{|\mathcal{H}|}}{\text{minimize}} \quad & f(\underline{x}) = \sum_{h\in\mathcal{H}} \frac{A_h^2}{x_h} \\ \text{subject to} \quad & \sum_{h\in\mathcal{H}} x_h = n \\ & m_h \le x_h \le M_h, \qquad h \in \mathcal{H}. \end{aligned}$$

2. Optimality conditions

   $\underline{x}^*$ is an optimal (regular) solution to Problem BC if and only if $\underline{x}^* = (x_h^{(\mathcal{L},\mathcal{U})},\ h \in \mathcal{H})$, where

$$x_h^{(\mathcal{L},\mathcal{U})} = \begin{cases} m_h, & h \in \mathcal{L} \\ M_h, & h \in \mathcal{U} \\ A_h\, s(\mathcal{L},\mathcal{U}) \in (m_h, M_h), & h \in (\mathcal{L}\cup\mathcal{U})^c, \end{cases}$$

   with $\mathcal{L}\cup\mathcal{U} \subsetneq \mathcal{H}$ such that

$$\mathcal{L} = \left\{ h \in \mathcal{H} :\ s(\mathcal{L},\mathcal{U}) \le \frac{m_h}{A_h} \right\}, \qquad \mathcal{U} = \left\{ h \in \mathcal{H} :\ s(\mathcal{L},\mathcal{U}) \ge \frac{M_h}{A_h} \right\}.$$

3. New *RNABOX* algorithm that solves Problem BC (and generalizes existing *RNA*).
4. R package `stratallo`.

References

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*, Cambridge University Press, Cambridge.

Cont, R. and Heidari, M. (2014). Optimal rounding under integer constraints.
https://arxiv.org/abs/1501.00014

Friedrich, U., Münnich, R., de Vries, S. and Wagner, M. (2015). Fast integer-valued algorithms for optimal allocations under constraints in stratified sampling, *Computational Statistics & Data Analysis*, 92, pp. 1–12.
https://www.sciencedirect.com/science/article/pii/S0167947315001413

Gabler, S., Ganninger, M. and Münnich, R. (2012). Optimal allocation of the sample size to strata under box constraints, *Metrika*, 75(2), pp. 151–161.
https://doi.org/10.1007/s00184-010-0319-3

Münnich, R. T., Sachs, E. W. and Wagner, M. (2012). Numerical solution of optimal allocation problems in stratified sampling under box constraints, *AStA Advances in Statistical Analysis*, 96(3), pp. 435–450.
https://doi.org/10.1007/s10182-011-0176-z

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

Wesołowski, J., Wieczorkowski, R. and Wójciak, W. (2021). Optimality of the Recursive Neyman Allocation, *Journal of Survey Statistics and Methodology*, 10(5), pp. 1263–1275.
https://academic.oup.com/jssam/article-pdf/10/5/1263/46878255/smab018.pdf
https://arxiv.org/abs/2304.07034

Wright, T. (2017). Exact optimal sample allocation: More efficient than Neyman, *Statistics & Probability Letters*, 129, pp. 50–57.
https://www.sciencedirect.com/science/article/pii/S0167715217301657

Wójciak, W., Wesołowski, J. and Wieczorkowski, R. (2023). R Package stratallo - source code.
https://github.com/wwojciech/stratallo

Wójciak, W. (2023). `stratallo`*: Optimum Sample Allocation in Stratified Sampling*. R package version 2.2.1.
https://CRAN.R-project.org/package=stratallo

✉ wojciech.wojciak.dokt@pw.edu.pl

Last updated: 2023-06-27