Sample Size Optimization of the Consumer Price Index: An Implementation using R

Harold Gomes and William Johnson
U.S. Bureau of Labor Statistics (BLS)
2 Massachusetts Ave NE, Room 3655, Washington, D.C. 20212

Abstract

The Consumer Price Index (CPI) is estimated based on a multistage probability sampling design. To collect the optimal number of Items and Outlets across the United States, a non-linear constrained optimization method, known as the *Item-Outlet Optimization Program* (IOOP), has been used. IOOP calculates optimal sample sizes for the commodities and services component of CPI, about 70% of the CPI weight. Previous BLS literature has described the mathematical basis of this method. Currently, CPI uses SAS for computation. In this study, we provide useful technical details for practical implementation in R, intuitive interpretation and infographics. What makes IOOP unique compared to classic methods, such as Neyman allocation, is its level of practicality and complexity. IOOP generates optimal sample sizes to minimize the overall CPI variance while maintaining fixed budgets and scope, as well as other constraints. The fixed scope is essentially the parameters—labor hours, travel time, response rates, etc.—to account for the reality of data collection. The R implementation provides a validation of SAS results, and the details can be beneficial to agencies seeking a method to account for scope.

Key Words: Consumer Price Index (CPI), nonlinear optimization, optimal sample size, Item Outlet Optimization Program (IOOP), optimal allocation, minimize variance

Any opinions expressed in this paper are those of the author and do not constitute policy of the Bureau of Labor Statistics.

1. Introduction

1.1 Historical Background

Since its 1978 revision, the Consumer Price Index (CPI) has been estimated through a scientific sampling method, a multistage probability sampling design. Meanwhile, statisticians at the Bureau of Labor Statistics (BLS) started to brainstorm and develop a theoretical framework for optimal sample allocation methodology for the commodities and services component of CPI (about 70% of the CPI weight; housing sampling comprises the remaining 30%). To collect the optimal number of *Items* and *Outlets* across the United States, a non-linear constrained optimization method, known as the *Item-Outlet Optimization Program* (IOOP), has been used. Richard Valliant, Sylvia Leaver, William Johnson, Owen Shoemaker, Thomas Benson, Darin Solk, Curtis Jacobs, William Weber, and Michael Cohen contributed in developing the mathematical basis and published literatures on mathematical derivations of IOOP design (Leaver et al. 1986, 1999, 2005; Johnson et al. 1999; Shoemaker et al. 1999). The first operational implementation of IOOP

(as a regular task) employed SAS PROC NLP (1999) for solution, followed by S-PLUS NuOPT to ensure that SAS and S-PLUS produce similar results. SAS IML was also implemented a few years after SAS NLP. Prior to regular implementation, other expired or customized programming languages were used during the development phase, field testing and model validation of the procedure. Currently, CPI uses the SAS OPTModel for computation. Technical details were not discussed much in previous literature, as the discussion typically encompassed the mathematical derivations of IOOP and the sample design of CPI.

1.2 Current Study

In this study, we take the opportunity to discuss the technical details, challenges and mitigation strategies (section 4) as we implement IOOP in R. These technical details may be beneficial to other statistical agencies or organizations that are trying to implement a large scale optimization model for sample size calculations. We also provide intuitive interpretations to promote the merit of IOOP in survey sampling estimation, and infographics for conceptual clarity of IOOP. Infographics and data visuals are becoming popular tools to communicate scientific findings and methodology to other scientists, researchers, stakeholders and executive leaders in the field of research and statistical product development.

2. What Does IOOP Really Do?

The principal research question that IOOP attempts to discover is:

How many *Outlets (M)* and *Items (1/K)*[1] should we collect (sample size) for an efficient overall CPI estimate (i.e., with minimum variance) across the United States while maintaining fixed multipurpose budgets and scope (i.e., labor hours, travel time, response rates, modes of collection, number of PSUs, etc.) and other constraints to account for the reality to operate a statistical survey program?

In this perspective, IOOP is a dynamic, complex, and practical sample size calculator to aid the field staff for price collection of *items*[2] from different *outlets* across the United States. Another benefit of this method is that it generates a model predicted CPI variance estimate every 6 months (commodities and services component (C&S), ~70%) based on the optimal sample sizes it generates as output. Additionally, this method provides simultaneous solutions to both, *item* and *outlet* sample sizes (Leaver et al. 1986), which is not typically observed in classic survey sample allocation design due to the design complexity and computational challenge. What makes IOOP unique compared to classic methods such as Neyman allocation is that IOOP accounts for many parameters including multipurpose budgets and differential costs, constraints, and scope that are part of the reality to operate a statistical survey program and complex data collection procedures. In other words, ignoring scope parameters in the design for the trade-off of simplicity may generate unachievable sample sizes. It is because the scope parameters are so vital in operating a statistical organization. Hence, IOOP attempts to capture the feasibility of a context prior to generating the optimal sample sizes for collection in order to achieve the best minimum U.S. level CPI standard error. Neyman allocation "ignores any differential costs of data collection and processing among strata" in addition to a fixed overall target sample size (Valliant et al. 2013).

[1] Optimization program outputs K's, which are then inversed (1/K) to produce the *Item* sample size.

[2] A specific type of *item* is known as *quote*. In this paper, *item* and *quote* are often used interchangeability for simplicity.

2.1 Nuts and Bolts of IOOP: Accounting Practicality and Complexity

Four key ingredients formulate an optimization problem—objective function, decision variables, parameters, and constraints (Valliant et al. 2013, Gonzalez et al. 2010). The goal of an optimization problem is to find (output) the optimal values for the decision variables that will enable the minimum (or maximum) value for the objective function while maintaining all the constraints. Parameters are the coefficients (constants) nested within the objective function, often properties or attributes that have functional relationship with the objective function or with a constraint function.

2.1.1 Objective function: minimize the variance

The objective function of IOOP is the projected variance $\sigma^2(PC_{total})$ for the 6-month percent price change of All Items less Shelter (C&S) of the U.S. CPI. The projected variance can be written as

$$\sigma^2(PC_{total}) = \sum_{area=1}^{38}\sum_{mg=1}^{13}\left(RI_{mg,area}^2 * \sigma^2\left(PC_{mg,area}\right)\right) = \sum_{pg=1}^{15}\sum_{mg=1}^{13}\left(RI_{mg,pg}^2 * \sigma^2\left(PC_{mg,pg}\right)\right)$$

The projected variance for a single index area or major group (mg) belonging to a PSU group (pg) can be written as

$$\sigma^2\left(PC_{mg,area}\right) = \sum_{mg=1}^{13} RI_{mg,area}^2\left(\frac{\sigma_{item,mg,area}^2}{f_1\left(M_{mg,pg},K_{mg,pg},N_{area}\right)} + \frac{\sigma_{outlet,mg,area}^2}{f_2\left(M_{mg,pg},K_{mg,pg},N_{area}\right)} + \frac{\sigma_{error,mg,area}^2}{f_3\left(M_{mg,pg},K_{mg,pg},N_{area}\right)} + \frac{\sigma_{psu,mg,area}^2}{f_4\left(M_{mg,pg},K_{mg,pg},N_{area}\right)}\right)$$

Where

$$f_1\left(M_{mg,pg},K_{mg,pg},N_{area}\right) = \frac{N_{area}}{K_{mg,pg}}$$

$$f_2\left(M_{mg,pg},K_{mg,pg},N_{area}\right) = NRO_{mg} * N_{area} * \left(\left(AV_{mg,area} + NPV_{mg}\right) * M_{mg,pg} + BV_{mg,area} * M_{mg,pg}^2\right)$$

$$f_3\left(M_{mg,pg},K_{mg,pg},N_{area}\right) = \frac{M_{mg,pg}}{K_{mg,pg}} * N_{area} * NRQV_{mg}$$

$$f_4\left(M_{mg,pg},K_{mg,pg},N_{area}\right) = N_{area}$$

In the following sections and in Table 1, we discuss the parameters in details. See Leaver et al. (1999), Johnson et al (1999, 2016) for the mathematical derivations.

2.1.2 Decision variables: output the optimal sample sizes

Sample sizes for *Items* ($1/K_{mg,\ area}$) and *Outlets* ($M_{mg,\ area}$) are the decision variables in this optimization problem. Because it optimizes two stages of sampling out of three stages, it generates $1/K_{mg,\ area}$ and $M_{mg,\ area}$. First stage, PSU structure, is fixed. There are 13 Major Groups (mg) of items across 38 Index Areas (area) in the United States (13 x 38 = 494). Since samples are collected from each major group and from each index area, IOOP simultaneously generates 494 sample sizes for *Items* ($1/K_{mg,\ area}$) and 494 sample sizes for *Outlets* ($M_{mg,\ area}$). Hence, IOOP estimates 988 optimal samples sizes (decision variables) as output. This large variable problem (2 x 13 x 38 = 988) can be reduced into a smaller variable problem (2 x 13 x 15 = 2 x 195 = 390) by grouping all the PSUs into 15 PSU Groups (pg) based on the similarity of relative importance (weights). We will discuss more about this process in the technical detail sections as IOOP solves for 390 decision variables.

2.1.3 Parameters: scope

Model parameters (coefficients of the decision variables) are what make IOOP a unique sample allocation problem. It enriches IOOP into a pragmatic sample allocation method since it accounts for the scope—parameters that are associated with operating a statistical survey program, data collection procedure, and outlet-item response rate behaviour. Table 1 provides all the parameters and associated scope within the context.

2.1.4.1 Non-linear constraint: realistic cost function

Another aspect of IOOP that is unique compared to many other allocation models, including classic models, is its handling of survey-associated costs as a non-linear constraint function. In practice, an overall budget is often itemized into multipurpose budgets as a function of office, purpose or other factors that are not controlled by a chief survey officer; or simply due to differential costs among strata or data processing. Additionally, there are fixed costs and variable costs in the production setting which adds complexity to survey sampling. IOOP captures this non-linear functional relationship of various costs. The non-linear cost function is the sum of four multipurpose cost functions—outlet related initiation costs ($CIO_{mg,pg}$), outlet related repricing costs ($CPO_{mg,pg}$), quote related initiation costs ($CIQ_{mg,pg}$), and quote related repricing costs ($CPQ_{mg,pg}$). Fixed and variable costs are accounted within each of the four cost functions. They are: compensation, per-diem, mileage, travel time, overlap adjustments, seasonal cost, and differential costs for survey modes of collection and response rates.

$$C_{total} = \sum_{mg=1}^{13} \sum_{pg=1}^{15} \Big(CIO\big(M_{mg,pg}, K_{mg,pg}\big) + CIQ\big(M_{mg,pg}, K_{mg,pg}\big) + CPO\big(M_{mg,pg}, K_{mg,pg}\big) + CPQ\big(M_{mg,pg}, K_{mg,pg}\big) \Big)$$

Where

$$CIO\big(M_{mg,pg}, K_{mg,pg}\big) = 0.25 * N_{pg} * \big(CO_{mg} + COT_{mg}\big) * areafactor * \Big(\big(AC_{mg,area} + NPC_{mg}\big) * M_{mg,pg} + BC_{mg,area} * M_{mg,pg}^2\Big)$$

$$CIQ\big(M_{mg,pg}, K_{mg,pg}\big) = 0.25 * N_{pg} * WOD_{mg} * CQ_{mg} * NRO_{mg} * \frac{M_{mg,pg}}{K_{mg,pg}}$$

$$CPO\big(M_{mg,pg}, K_{mg,pg}\big) = N_{pg} * MBO_{mg,pg} * NRO_{mg} * \Big(\big(AC_{mg,area} + NPC_{mg}\big) * M_{mg,pg} + BC_{mg,area} * M_{mg,pg}^2\Big) * \Big(\big(CPVO_{mg} + CPO_{mg}\big) * RPVO_{mg} * areafactor + CTO_{mg} * RTO_{mg} + CWO_{mg} * RWO_{mg}\Big)$$

$$CPQ\big(M_{mg,pg}, K_{mg,pg}\big) = N_{pg} * MBQ_{mg,pg} * NRQC_{mg} * \frac{M_{mg,pg}}{K_{mg,pg}} * \big(CPVQ_{mg} * RPVQ_{mg} + CTQ_{mg} * RTQ_{mg} + CWQ_{mg} * RWQ_{mg}\big)$$

Table 1 provides the details about the parameters.

2.1.4.2 Linear constraints: bounds and restrictions for sample sizes

Maximum or minimum limits on the decision variables (sample sizes) constitute the linear constraints of IOOP. There are a total of 195 (13 Major Group x 15 PSU Group) linear constraints that restrict the Items and Outlets sample sizes, and 1 linear constraint function. A study by Bradley (2005) suggested that small sample size from the lowest sampling unit level (area-item) induces an upward bias in the CPI-U measure, and proposed a small sample bias adjustment factor by estimating the second order stochastic expansion of the index. Hence, IOOP employs a linear function to restrict the output sample size that accounts for the small area sample bias in the design level. There are two

Table 1: Optimization Parameters (model coefficients) that formulate IOOP. They account for the reality to operate a statistical survey program and data collection procedure (scope)

	Scope	Parameter	Notation
1	Item Strata or Area Structure	number of PSUs in the index area (referred to as "by area")	N_{area}
2	Item Strata or Area Structure	weighted sum of nonpops categories in major group (mg)	NPV_{mg}
3	Item Strata or Area Structure	number of PSUs in PSU group pg (as opposed to the number of PSUs in an index area referred to as area)	N_{pg}
4	Item Strata or Area Structure	minimum number of non-self representing PSUs in the PSU group (pg)	N_{nsr}
5	Item Strata or Area Structure	number of item strata in major group (mg)	$N_{is,mg}$
6	Labor Hours	compensation initiation cost per outlet (mg)	CO_{mg}
7	Labor Hours	compensation cost for a personal visit for pricing per outlet (mg)	$CPVO_{mg}$
8	Maximum Grand Cost	Cost ceiling for initiation and repricing for 1 year	C_{total}
9	Modes of Collection *(Differential Cost* or *Diff Cost)*	cost of telephone collection of an outlet (mg)	CTO_{mg}
10	Modes of Collection; Diff Cost	cost of internet collection of an outlet (mg)	CWO_{mg}
11	Modes of Collection; Diff Cost	per quote cost for a personal visit for pricing (mg)	$CPVQ_{mg}$
12	Modes of Collection; Diff Cost	per quote cost of telephone collection (mg)	CTQ_{mg}
13	Modes of Collection; Diff Cost	per quote cost of internet collection (mg)	CWQ_{mg}
14	Modes of Collection; Diff Cost	percent of quotes collected by personal visit (mg)	$RPVQ_{mg}$
15	Modes of Collection; Diff Cost	percent of quotes collected by telephone (mg)	RTQ_{mg}
16	Modes of Collection; Diff Cost	percent of quotes collected by internet (mg)	RWQ_{mg}
17	Multipurpose Budgets	outlet related initiation costs	$CIO_{mg,pg}$
18	Multipurpose Budgets	outlet related repricing costs	$CPO_{mg,pg}$
19	Multipurpose Budgets	quote related initiation costs	$CIQ_{mg,pg}$
20	Multipurpose Budgets	quote related repricing costs	$CPQ_{mg,pg}$
21	Multipurpose Budgets	initiation cost per quote (mg)	CQ_{mg}
22	Overlap Element Adjustment or Non-Duplication	linear and quadratic coefficients of the unique outlet predictor function for variance projection	$AV_{mg,area}$; $BV_{mg,area}$
23	Overlap Element Adjustment or Non-Duplication	linear and quadratic coefficients of the unique outlet predictor function for cost	$AC_{mg,area}$; $BC_{mg,area}$
24	Overlap Element Adjustment or Non-Duplication	factor to adjust for the monthly/bimonthly mix of outlets for PSU group (pg) and major group (mg)	$MBO_{mg,pg}$
25	Response Rate	outlet level response rate (for major group, mg)	NRO_{mg}
26	Response Rate	quote level response rate (mg)	$NRQV_{mg}$
27	Response Rate	percent of outlets collected by personal visit (mg)	$RPVO_{mg}$
28	Response Rate	percent of outlets collected by telephone (mg)	RTO_{mg}
29	Response Rate	percent of outlets collected by internet (mg)	RWO_{mg}
30	Response Rate	quote level response rate for projected costs (mg)	$NRQC_{mg}$
31	Seasonal Cost	seasonal items initiation factor (mg)	WOD_{mg}
32	Small Sample Bias	average number of quotes in the lowest level sampling unit (area–item)	SSB
33	Travel Distance or Time	perdiem and mileage cost per outlet (mg)	COT_{mg}
34	Travel Distance or Time	travel cost for a personal visit for pricing per outlet (mg)	CPO_{mg}
35	Travel Distance or Time	Ratio of current expected distance to nearest neighbor divided by 1987 distance to nearest neighbor	areafactor

slightly different linear equations depending on *self* or *non-self* representing PSU groups. These constraints specify that the average number of quotes should be greater than or equal to nine in the lowest level sampling unit (area–item) (or mg-pg).

For *self* representing PSU groups

$$9 * K_{mg,pg} - NRQC_{mg} * \frac{1}{N_{is,mg}} * M_{mg,pg} \leq 0$$

For *non-self* representing PSU groups

$$9 * K_{mg,pg} - \frac{1}{3 - MBQ_{mg,pg}/6} * NRQC_{mg} * \frac{N_{nsr}}{N_{is,mg}} * M_{mg,pg} \leq 0$$

3. Optimization Infographics

Infographics are designed to promote the conceptual clarity of IOOP, and to illustrate the complexity of the optimization process. Figure 1 illustrates the IOOP process, and Figure 3 illustrates the disaggregation process for sample size.

4. Useful Technical Notes: Challenges and Mitigation

In scientific computing and algorithmic perspective, optimization *design* can be classified into four increasing levels of complexity—unconstrained, linear program, quadratic program, and non-linear program optimizations. IOOP is the most complex in this pyramid. On the other hand, an optimization *algorithm* can be classified into three main categories—analytic gradient based, finite-difference gradient based and non-gradient based techniques. Figure 2 (Optimization Designs) provides an infographic to explain the properties and trade-offs of these algorithms.

The first operational implementation of IOOP (1999) employed Windows 95 operating system with SAS NLP. Prior to this, a mainframe computer environment was used during model development and validation. Fortunately, computing power is no longer a challenge in 2016 as it was when IOOP was first implemented, which gives rise to many creditable algorithm packages for linear and non-linear optimization in R. To name a few CRAN verified packages: nloptr, Alabama, NlcOptim, mopsocd. Additionally, the availability of RAM (random access memory)—in order to load, store, and use temporary data—is no longer a challenge as it was a decade ago in terms of cost, quality and quantity.

Another property of IOOP that's worthy of mention is its improvement to variance component estimation methodology (Shoemaker, 1999) by introducing a restricted maximum likelihood (REML) in order to reduce the effect of nuisance information. This improvement resulted in stability of variance components and the all-aggregate variance.

We implemented an analytic gradient (partial derivatives) based approach in R using *nloptr* package. *nloptr* is an interface between R and the NLopt open source library of optimization algorithms[3]. Authors of this package also demonstrate their innovative research work with optimization algorithms[4].The specific algorithm used is the *Sequential Least Squares Quadratic Programming* method, referred to as

[3] Steven G. Johnson, The NLopt nonlinear-optimization package, http://ab-initio.mit.edu/nlopt

[4] Eddelbuettel, D. (2013). Seamless R and C integration with Rcpp. New York: Springer.

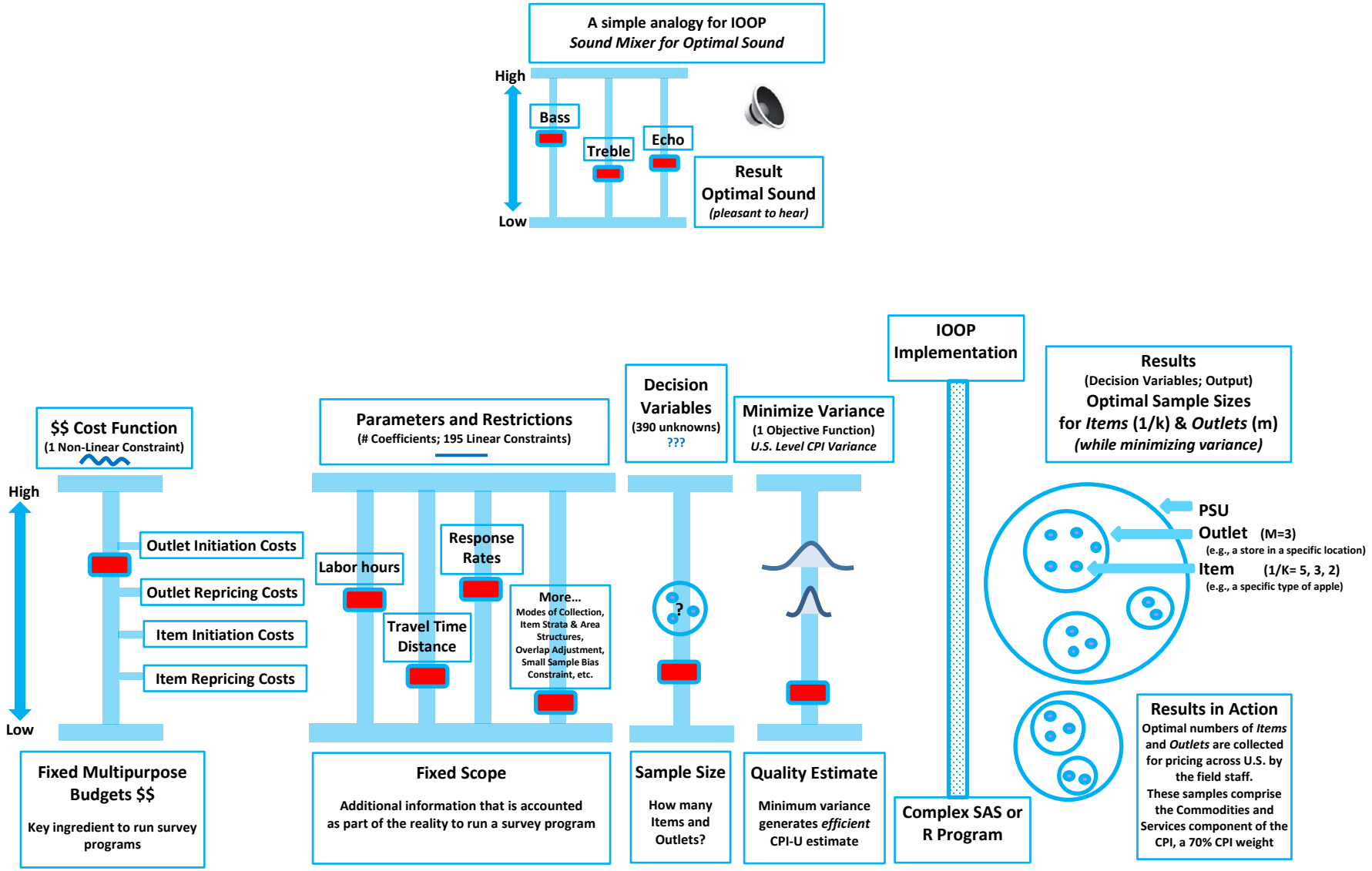

Figure 1: Infographic of *Item Outlet Optimization Program* (IOOP)

NLOPT_LD_SLSQP, which can handle non-linear objective functions, and non-linear equality and inequality constraints. This algorithm allows the user to specify the analytic gradient of the objective function and the Jacobians of the equality and inequality constraints that greatly speeds up the algorithm and allows it to converge to a better point with less violation of any of the constraints. An advantage of R over other proprietary software is that the original source code algorithm by authors can be viewed and stored for assessing the details of a method.

4.1 Challenges and Mitigation for IOOP

4.1.1 What are the Challenges (C)?

(C1) IOOP is a large scale non-linear optimization program (Figure 2), thus very complex and requires a large amount of computing power.

(C2) The presence of many constraints (196), linear (195) and non-linear constraints (1), and pre-calculated coefficients (model parameters) tremendously shrink the *feasible region* for solution space i.e., sample sizes. For example, a rectangle full of feasible points generated by the upper and lower bounds (4 points) would be reduced simply by a line—formed by a linear constraint equation of K and M—passing through this rectangle. Now, shrink this space more by adding 195 of them! Here is an intuitive example: assuming each pair of K and M (a linear constraint equation line) reduces the feasible region of the plane by ½, the total feasible region is thereby reduced to (½) (½)…. (½) = $(½)^{195} = 2 \times 10^{-59}$ in this manifold (topological space). In plain language, it means the cross products of 195 planes tremendously reduce the feasible region in IOOP.

(C3) Objective function values at neighbouring points are too close to differentiate the *best value* in convergence within machine precision. In algorithmic perspective, converging in a single or two iterations may indicate this ill-conditioned behaviour.

(C4) Item sample size allocated to each index area in a PSU group and in a major group $K_{mg,pg}$ is a *non-linear* function of the minimum sample size bias constraint (≥ 9). Hence, it increases the complexity by introducing additional 195 (15x13) non-linear constraints.

$$NRQC_{mg} * \frac{1}{N_{is,mg}} * \left(K_{mg,pg} * M_{mg,pg}\right) \geq 9 \qquad (non-linear\ constraint)$$

4.1.2 What are the Mitigation (M) Strategies?

(M1) Sufficient RAM and quality computer processors (CPUs) can mitigate C1, although longer time may still be a challenge. Gradient based optimization may mitigate the time challenge as discussed in the next section.

(M2) Shrinking of *feasible region* may result in "no feasible solution" or non-convergence errors, or simply the algorithm may not run or continue. To mitigate C2, initial points (988) must be a set of points that are very close to the *feasible region* in order for the optimization algorithm to iterate and to find the next best set of points. In practice, using the solution set from previous rotation as the initial starting points for the following rotation mitigates this challenge. The constraint precision (tolerance level) may also need to be relaxed to avoid "no feasible solution" errors (discussed more in the next section).

(M3) Geometrically, raising the power ($n\ for\ f(x) > 1\ ; \frac{1}{n}\ for\ f(x) < 1$) of a positive valued function often increases the curvature of the function that results in exaggeration of the distance between clustered points. To mitigate C3, the overall objective function is raised to the $1/5^{th}$ in order to *exaggerate* the difference at nearby point values. It results in converging to the *best* possible point (minimum variance) with the corresponding optimal output solution points (sample sizes). The premise to mitigate this challenge is the *monotonically increasing* property of our objective function (within given bounds). If a function $f(x_i)$ is *monotonically increasing*, then a set of points (x_i) that achieves the

minimum value for $g(x_i)$ is the *same-set* of points (x_i) that achieves the minimum for the transformed function $f(g(x_i))$. In other words, the decision variables (sample sizes) are invariant with respect to minimum values of the original objective and root transformed objective functions due to *monotonically increasing* property.

(M4) Dividing both sides of C4 equation by item sample size $(1/\ K_{mg,pg})$, 195 *non-linear* constraints could be expressed as *linear* constraints (eliminating product of $K_{mg,pg} * M_{mg,pg}$) enabling simplicity for IOOP computation. In this way, IOOP constitutes 195 linear constraints and 1 non-linear constraint (cost function) instead of 196 non-linear constraints.

$$NRQC_{mg} * \frac{1}{N_{is,mg}} * M_{mg,pg} \geq 9 * \frac{1}{K_{mg,pg}} \qquad (linear\ constraint; N_{is,mg}\ is\ constant)$$

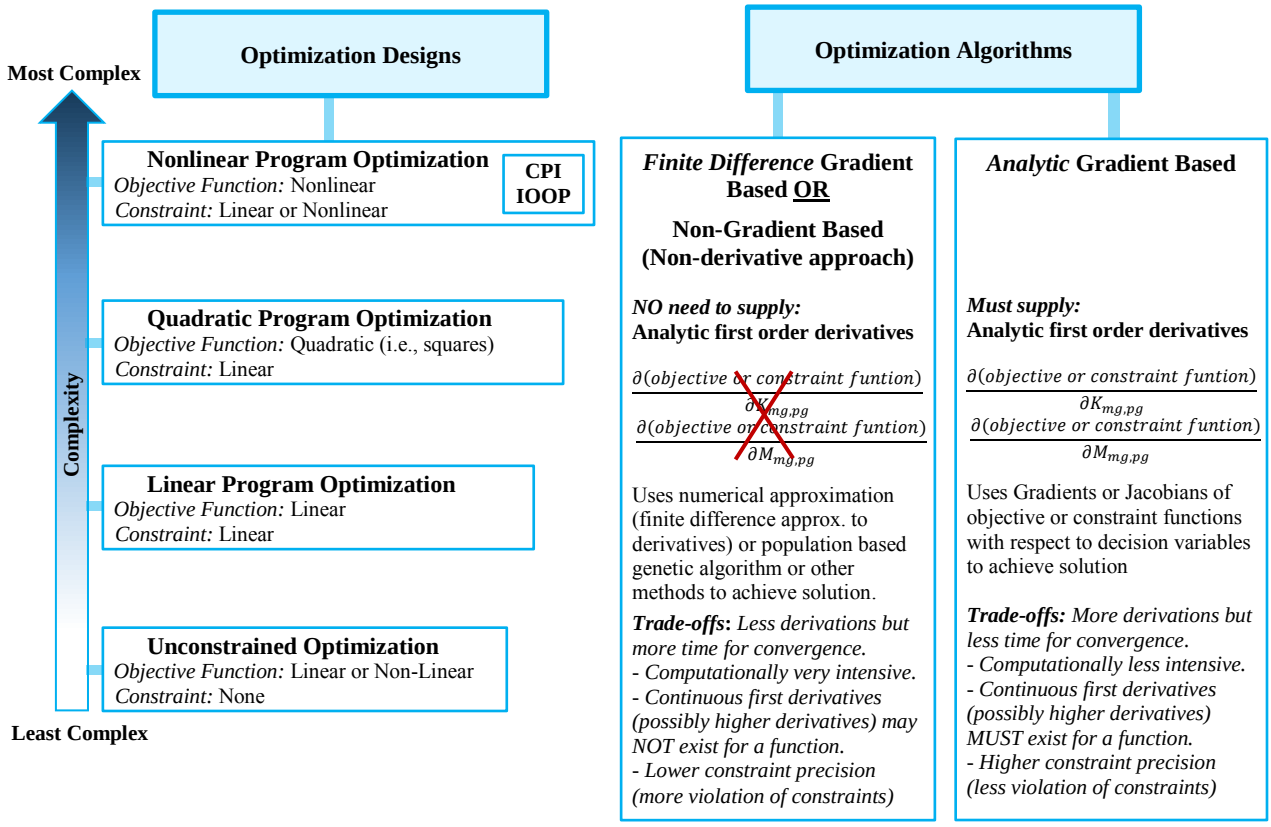

Figure 2: Optimization problem classification, in *design* and in *algorithm*, based on scientific computing and algorithmic perspectives.

4.2 Trade-Offs in *Analytic* and *Finite Difference* Based Gradient Algorithms

The gradient of a function (1st order derivatives) provides the *best* direction (steepest descent) to travel in order to find the minimum or maximum of a function. Optimization algorithms can be classified into gradient and non-gradient based approaches, and within the gradient approach, analytic and finite difference based algorithms. An infographic—Optimization Algorithm (Figure 2)—is designed to clarify the properties and trade-offs in each method. One major benefit to an analytic gradient based method in practice is that it

takes much less time to converge due to presence of analytic partial derivatives, and it converges to a better point with less violation of any of the constraints (i.e. higher constraint precision or tolerance level). In this algorithm, the analytic gradient is evaluated in each iteration step faster than the finite difference based algorithm. The evaluation of the gradient continues until the algorithm achieves its optimal solution points where it no longer can descend to minimum or maximum direction. The stopping rule for convergence is a trade-off of a few criteria, such as, absolute and relative differences of the objective function from one iteration to the next, absolute values of constraint violations, and absolute and relative differences of the values of decision variables from one iteration to the next. SAS OPTModel and SAS NLP do not require the user to supply any analytic gradient. In R, both options are available depending on the package. Fortunately, the IOOP functions are continuous and differentiable in first and second order derivatives, which allowed us to employ the analytic gradient based approach.

4.3 Challenges and Mitigation in *Implementation*

4.3.1 What are the Challenges (C)?

(C5) Constraint precision i.e., tolerance for constraint violation may often pose a challenge that results in "no feasible solution" errors. The feasible region is a very narrow space in IOOP due to 196 constraints.
(C6) Loops in the R program may pose a runtime challenge, as loops may take a longer time to execute and complete all of the algorithmic steps.
(C7) IOOP not only has one function but multiple functions to be implemented.
(C8) Partial derivatives for Gradient and Jacobean need to be supplied in order to speed up the computation process using the analytic gradient based approach.
(C9) Computing a large number of decision variables (988) may be a challenge for performance time in a production context, as it was the case in the 1999 implementation (ran for a week including day and night).

4.3.2 What are the Mitigation (M) Strategies?

(M5) Having a control option to experiment with different tolerance level mitigates this challenge. A too high precision level may result in not finding any solution, while a too low precision level may result in "premature" convergence without the best value for objective function (true minimum). R/Nloptr has control options for tolerances. The analytic gradient based approach in this implementation indeed provides some advantage for higher constraint precision. Table 2 shows tolerance level for this implementation.
(M6) R is a function friendly language. Functions can run many times faster than a Loop (Uyttendaele, 2015). Thus, avoiding loops wherever possible and using vector based formulas within a function is a wise choice for faster performance.
(M7) All the functions—objective, non-linear constraint, linear sample bias constraint—need to be coded in R, and R/Nloptr has options for all of the functions to be incorporated.
(M8) Partial derivatives (Gradient and Jacobean) need to be mathematically derived for the objective function and non-linear cost function before coding them in R.
(M9) To get around the severe performance issues with 988 decision variables in 1999, the problem was simplified by clustering index areas into PSU groups and giving the same sample design to each index area in the PSU group. This is no longer necessary and at some point in the future, IOOP will no longer use PSU groups. Indexing technique in coding an algorithm is a useful strategy to mitigate this challenge, as indexing enables using large coefficient datasets, clustering them into smaller vectors, running them through the loops, and translating the output back to the original dataset size. That is, the indexing strategy enables one to turn a large variable problem (2 x 13 x 38 = 988) into a reduced

variable problem (2 x 13 x 15 = 2 x 195 = 390) when performance time for production or computing machine creates a technical challenge.

5. Model Validation using Dataset: R and SAS Solution Comparison

In a statistical survey program, validation using another environment is often a good practice to preserve the reliability of solutions or for a backup plan. This R implementation provides a validation of SAS results using another computing environment. During the developing, testing and validation phases of the R IOOP program, we used actual datasets to compare the SAS and R solutions—August 2015 and February 2016 IOOP cycles. Table 2 summarizes the results for February 2016. R/nloptr produced a slightly better (smaller) objective function value, although the difference is so small (0.00889958 %) that it makes no practical difference. The final objective function for SAS violated the cost ceiling by 0.042 for February 2016 dataset, while R violated by less than 10^{-7}. The maximum violation of any of the small sample size constraints was also less than 10^{-7} for R.

Table 2: February 2016 IOOP results comparison of SAS and R solutions

Software	*Objective Function (projected minimum variance)*	*Tolerance Level (cost ceiling violation)*	*Tolerance Level (small sample size constraint violation)*
SAS/OPTModel	0.05045631	0.042	$< 10^{-6}$
R/nloptr	0.05045182	$< 10^{-7}$	$< 10^{-7}$

To assess the 390 (2 x 13mg x 15pg) sample sizes of Items and Outlets between SAS and R solutions, we plotted the *difference* in corresponding outlet sample sizes $\{M_{mg,pg}(R) - M_{mg,pg}(SAS)\}$ and in corresponding item sample sizes $\{1/K_{mg,pg}(R) - 1/K_{mg,pg}(SAS)\}$ for the February 2016 projection (Figure 3). (Decimals are allowed in optimization since solution points are treated as continuous. Transforming decimal sample sizes into integer sample sizes is a *post-hoc* processing after solving the IOOP design problem).

In summary, this comparison indicates that the optimal sample sizes found by R and SAS are very similar, and validates the R implementation of IOOP Program (Figure 3; Table 2).

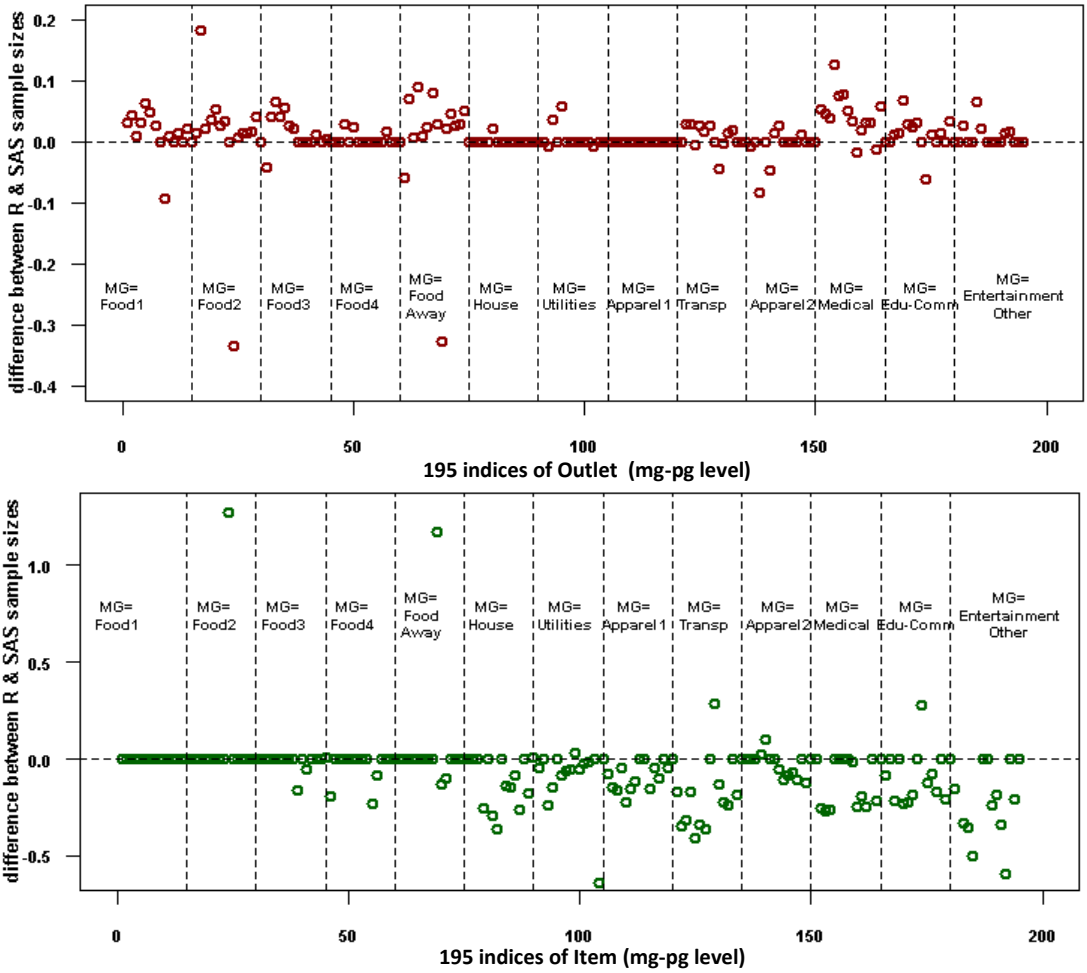

Figure 3: *Graphical Diagnosis*—Assessing the sample size difference between R and SAS solutions for 195 indices for February 2016 dataset; 195 mg-pg of Outlets and 195 mg-pg of Items.

5.1 From Optimization Design to Post-Hoc Processing

The IOOP *design* problem ends once it outputs the sample sizes, followed by *post-hoc* processing. CPI has 211 Item Strata, 8 Major Groups[5], and 38 Index Areas. Solving 170 x 38 design[6] problem (as opposed to 13 x 15) would have been ideal, as it implies to compute 6,460 decision variables (sample sizes) in the optimization method. Thanks to statistical science and its utility of homogeneity principal (similarity) across various contexts including IOOP, as it has been reduced to a problem of 13 Major Groups x 15 PSU Groups, i.e., 195 decision variables. This is where the *post-hoc* processing comes in handy to disaggregate 195 sample sizes into 6,460 sample sizes that are the building blocks of the CPI (8,018 basic item-area index all inclusive[6]). An infographic is designed to provide clarity of this process (Figure 4). Hence, IOOP essentially generates *aggregate* sample sizes that are disaggregated in post-hoc processing using various methodologies.

[5] Based on similarities of elements, IOOP design uses 13 Major Group (i.e., Item Strata Groups) instead of 8.

[6] IOOP generates sample sizes for 170 Item Strata although CPI has a total of 211 Item Strata that include 31 unpriced, 2 rent, and 8 fixed-design priced strata.

$M_{mg,pg}$— To be precise, the number of outlet hits allocated per POPS[7] category for major group mg and PSU group pg—is rounded into integer sample size value using *randomized rounding* while preserving the expectation value (mean) over the entire $M_{mg,pg}$ sample space.

$1/K_{mg,pg}$— To be precise, the number of item stratum selections allocated to each index area in PSU group pg and major group mg—requires few more steps of post-hoc processing. Item strata group ($1/K_{mg,pg}$) sample size is first disaggregated into Item

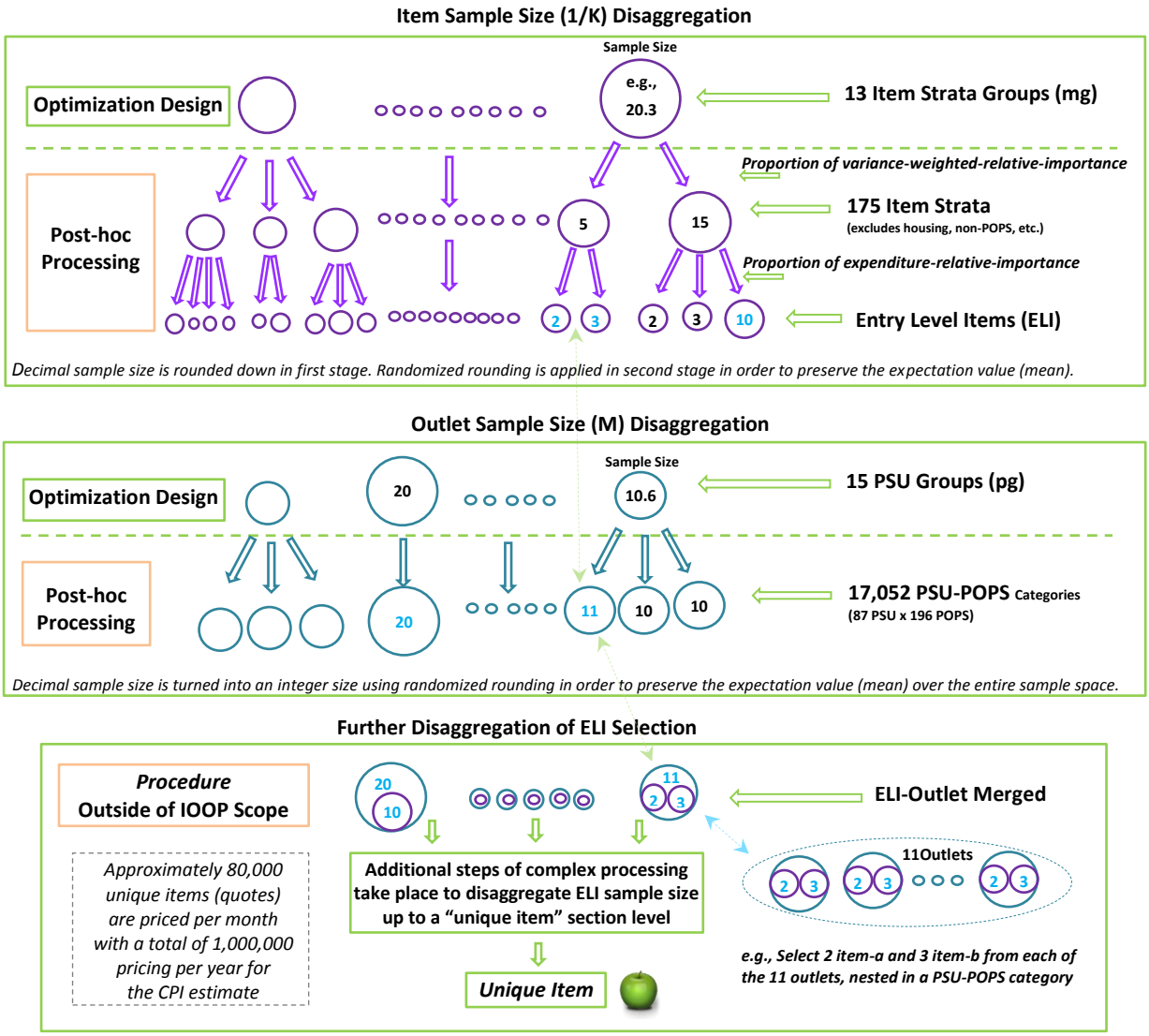

Figure 4: *Infographics of sample size processing*— they display how the sample size output from the optimization *design* phase enters into a *post-hoc* processing phase for final sample size estimation of Outlets and Items, followed by a few complex steps of processing before a "unique item" is selected for pricing.

[7] POPS (TOPS + Non-POPS) is the frame for the universe of outlets.

Strata and then into entry-level items (ELI or quotes) based on the proportion of variance-weighted-relative-importance. This technique ensures allocation of more sample units to items that display high variance and high relative importance in order to reduce the overall variance; i.e., it accounts for a trade-off between variance per item and the weight of the item. In this post processing, decimal sample sizes are rounded down and the left-over sample size (cumulative remainder) is re-distributed based on the proportion of variance-weighted-relative-importance.

There are multiple stages of disaggregation methods applied before a unique item is selected from the IOOP output of aggregate sample sizes. It is a complex procedure outside the scope of the IOOP design problem that employs various allocation methods in different stages of disaggregation, such as proportional, rank based, shelf space based, equal probability based, etc. (Figure 4).

6. Conclusions

6.1 Benefits to Other Surveys and Statistical Agencies

In this study, we have successfully implemented IOOP model in R for a multistage probability sampling—the Consumer Price Index. It provides a validation for SAS OPTModel results using another computing environment. Classic allocation methods, such as Neyman allocation, deliver a high quality technique that minimizes the variance for a fixed study budget and a fixed overall sample size (n), and estimates the optimal strata sample sizes (n_h) (Valliant et al. 2013). However, limitations in accounting many scope factors in Neyman allocation (e.g., differential costs (Valliant et al. 2013)) that are unavoidable for statistical survey programs foster a need for a more pragmatic, complex, and useful model, utilizing mathematical programming. IOOP not only accounts for multipurpose budgets, differential costs and the non-linear behaviour of expenditures (fixed and variable cost), but also accounts for a range of relevant factors—labor hours, travel time and distance, response rate, etc.—including small sample size bias adjustments. Additionally, limitations on a closed-form solutions to a complex multistage design is another challenge that promotes the need for mathematical programming—linear or non-linear optimization (Green, J., 2000). Hence, IOOP generates a timely updated optimization process as computing power no longer necessitates a simplistic model. The infographics (visualizations) of the optimization process foster an effective communication among the stakeholders and executive leaders to promote the benefit of a method, and they may be used as prototypes for other agencies and surveys. The R code can also be used as a prototype and customized for other surveys. It also demonstrates how *paradata* (e.g., labor hours, travel time) from data collection could be utilized in a survey program (Wagner et al., 2012) to estimate the optimal sample size. Last and not least, IOOP offers a versatility such that, as agency budget proposals (e.g., ± 10%) or other factors change, model coefficients (parameters) could be recalculated as many times as needed, which enable the option for generating multiple projected plans in advance for each circumstance. The final implementation could execute the plan that is within scope for a survey program.

References

Bradley, R. (2007). Analytical Bias Reduction for Small Samples in the U.S. Consumer Price Index. *Journal of Business & Economic Statistics*, Vol. 25, pp 337-346.

Bureau of Labor Statistics. (2015). Appendix 7: Sample Allocation Methodology for Commodities and Services. *Chapter 17: The Consumer Price Index of BLS Handbook of Methods.* Retrieved from http://www.bls.gov/cpi/cpi_methods.htm.

Eddelbuettel, D. (2013). *Seamless R and C integration with Rcpp*. New York: Springer.

Green, J. (2000). Mathematical Programming for Sample Design and Allocation Problems. *In JSM Proceedings of the Survey Research Methods Section*. Alexandria, VA: American Statistical Association. 688 – 692. Retrieved from http://www.amstat.org/sections/srms/proceedings/papers/2000_115.pdf.

Gonzalez, Jeffrey, and Eltinge, John. (2010). Optimal Survey Design: A Review. *In JSM Proceedings of the Survey Research Methods Section*. Alexandria, VA: American Statistical Association. Retrieved from http://www.bls.gov/osmr/pdf/st100270.pdf.

Jelmer, Y., Borchers, H., and Dirk, E. (2015). *nloptr: R interface to NLopt*. R package version 1.0.4.

Johnson, William H., Leaver, S.G., and Benson, T.S (1999). Modeling the Realized Outlet Sample for the Commodities and Services Component of the U.S. Consumer Price Index. *In JSM Proceedings of the Government Statistics Section*. Alexandria, VA: American Statistical Association, pp 304-308.

Johnson, William and Gomes, Harold. Implementing the IOOP Non-linear Programming Problem in R. *Memorandum*. U.S. Bureau of Labor Statistics: Washington DC. January 6, 2016.

Leaver, Sylvia, Weber, W., Cohen, M., and Archer, K. (1986). Determining an Optimal Item-Outlet Sample Design for the 1987 U.S. Consumer Price Index Revision. *In JSM Proceedings of the Survey Research Methods Section*. Alexandria, VA: American Statistical Association. 499 – 504. Retrieved from http://www.amstat.org/sections/srms/Proceedings/y2012/Files/305214_74459.pdf.

Leaver, Sylvia G., Johnson, William H., Shoemaker, Owen J. and Benson, Thomas S. (1999). Sample Redesign for the Introduction of the Telephone Point of Purchase Survey Frames in the Commodities and Services Component of the U.S. Consumer Price Index. *In JSM Proceedings of the Government Statistics and Social Statistics Sections*. Alexandria, VA: American Statistical Association. 292-297.

Leaver, Sylvia G., and Solk, D.T. (August, 2005). Handling Program Constraints in the Sample Design for the Commodities and Services Component of the U.S. Consumer Price Index. *In JSM Proceedings of the Survey Research Methods Section*. Alexandria, VA: American Statistical Association. Retrieved from http://www.amstat.org/sections/srms/Proceedings/y2005/Files/JSM2005-000866.pdf.

Optimization Model Basics: Mathematics Library User's Guide. (n.d.). Retrieved Jan. 14, 2016, from http://www.extremeoptimization.com/Documentation/Mathematics/Optimization/Optimization_Model_Basics.aspx.

Shoemaker, Owen, and Johnson, William. (1999). Estimation of Variance Components of U.S. CPI Sample Design 1998-1999. *Memorandum*. U.S. Bureau of Labor Statistics: Washington DC.

Uyttendaele, N. (March, 2015). How to speed up R code: An introduction. Retrieved April 20, 2016, from http://arxiv.org/abs/1503.00855.

Valliant R, Dever J, Kreuter F. (2013). *Practical Tools for Designing and Weighting Survey Samples*. New York: Springer.

Wagner, J., West, T., Kirgis, N., Lepkowski, J., Axinn, W., Ndiaye, S. (2012). Use of Paradata in a Responsive Design Framework to Manage a Field Data Collection. *Journal of Official Statistics*, Vol.28, No.4, 477-499.