Optimal Pairing for Stratum Collapse Methods with Interviewer-Level Measurement Error Shared Across Strata August 2005

Moon J. Cho and John L. Eltinge
U.S. Bureau of Labor Statistics
Moon J. Cho, 2 Massachusetts Avenue NE, Washington, DC 20212
(Cho.Moon@bls.gov)

Key Words: Collapse method; Consumer Expenditure Interview Survey; Interviewer effects; Nonsampling error; Reporting error; Stratified multistage sample design.

1. Introduction: Variance Estimation and Interviewer Assignment for the U.S. Consumer Expenditure Interview Survey

For some general background on the Consumer Expenditure (CE) Interview Survey and variance estimation therein, see Bureau of Labor Statistics (1997) and Eltinge, Cho and Lahiri (2005). In the current paper, four points are of special interest.

First, the CE Interview Survey is viewed primarily to produce estimates of mean expenditure per consumer unit (CU) within specified expenditure categories, and ratios of these means are used in calculation of the U.S. Consumer Price Index (CPI).

Second, the CE Survey uses a stratified multistage probability sample of households which represents the total U.S. civilian noninstitutional population. To select a representative sample of the population, the CE Survey divides the nation into many areas and then selects some of these areas, and the selected area is called a "Primary Sampling Unit" (PSU). The PSUs are groups of counties, or independent cities. There are self-representing PSUs and non self-representing PSUs. The self-representing PSUs are from metropolitan areas.There are 105 PSUs in our CE interview data. The CE Interview Survey collapsed 105 PSUs to form 80 variance PSUs and then assigned two variance PSUs to each variance stratum. The set of sample PSU's used for the Survey consists of 101 areas; from which 87 urban areas were selected by BLS for the CPI program (BLS Handbook, 1997, p.163). Within each selected PSU, a given sample CU, roughly equivalent to a household, is randomly assigned to one of two modes of data collection: interview or diary. The remainder of this paper will consider only data from the CE Interview Survey. The CE Interview Survey includes rotating panels: each CU in the sample is interviewed every 3 months over five calendar quarters and then is dropped from the survey. Each quarter, approximately 20 percent of the addresses are new to the Survey. The interviewer uses a structured questionnaire to collect both demographic and expenditure data in the Interview Survey.

Third, variance estimates are obtained by the BRR method. Under a standard design with two PSUs selected with replacement from each stratum, the standard BRR method selects one PSU from each stratum in a balanced manner to form a set of half samples. These half samples are used to compute the resulting variance estimator.

Fourth, in the traditional sampling literature, justification for this approach uses the (approximate) independence of sample selection across strata and PSUs, and focuses only on the sampling error component of survey error. However, in the CE Interview Survey, we often need to have variance estimators that account for both sampling and measurement error. In addition, interviewers often collect data in more than one PSU. For some variables, the interviewer-level component of measurement error may be nontrivial. Consequently, one must consider the modification of standard replicate-based variance estimators that will account appropriately for the correlation across strata and PSUs induced by interviewer level measurement error. This paper considers some simple variance estimators based on a collapsed-stratum approach. The collapse procedure is intended to ensure that the newly paired variance-PSUs do not share a common interviewer but have similar population characteristics. Specific matching algorithms are developed and applied to data from the CE Interview Survey. These algorithms arise from optimality criteria related to the bias and stability of the variance estimator, and use stratum and primary-unit level variables like population size, as well as interviewer characteristics.

The views expressed in this paper are those of the authors and do not necessarily reflect the policies of the U.S. Bureau of Labor Statistics.

2. Variance Estimators Based on Standard and Modified Pairings of Variance PSUs

In keeping with common practice, work with variance estimation for the CE Survey has tended to focus primarily on the variances of sampling errors as such. Consider a survey variable Y_{hij} for CU j in design PSU i in design stratum h and define the population total

$$Y = \sum_{h=1}^{L} Y_h$$

where $Y_h = \sum_{i=1}^{N_h} Y_{hi},\ Y_{hi} = \sum_{j=1}^{M_{hi}} Y_{hij}$, L is the number of design strata, and N_h is the number of design PSUs in design stratum h and M_{hi} is the number of CUs in design PSU i in design stratum h. Given probability weights w_{hij} and a two-PSU-per-stratum design, a simple estimator of Y is $\hat{Y} = \sum_{h=1}^{L} \hat{Y}_h$ where $\hat{Y}_h = \hat{Y}_{h1} + \hat{Y}_{h2}$ and $\hat{Y}_{hi} = \sum_{j \in S_{hi}} w_{hij} \hat{Y}_{hij}$. In addition, a simple estimator of the design variance of $\hat{Y}$ is

$$\hat{V}(\hat{Y}) = \sum_{h=1}^{L} \hat{V}(\hat{Y}_h) \ . \qquad (1)$$

where

$$\hat{V}(\hat{Y}_h) = \left(\hat{Y}_{h1} - \hat{Y}_{h2}\right)^2 \qquad (2)$$

Under the assumption that sampling is independent across strata and that design PSUs are selected with replacement, $\hat{V}(\hat{Y})$ is unbiased for the design variance of $\hat{Y}$. Under without-replacement sampling of PSUs and additional conditions, $\hat{V}(\hat{Y})$ will be a conservative variance estimator, due to omission of the finite population correction.

However, we sometimes wish to have an estimator of variance that accounts for both sampling and nonsampling errors. Under the conditions described in Section 1, the estimator (1) may have a negative bias if a given interviewer collects data in both PSUs selected from a given stratum h. Consequently, one may wish to replace the variance estimator (1) with

$$\hat{V}^*(\hat{Y}) = \sum_{g=1}^{G} \hat{V}^*(\hat{Y}_g) \qquad (3)$$

where the $\{2 \times L\}$ design PSUs are partitioned into G groups called "variance strata"; variance stratum g contains $n_{(g)}$ design PSUs; these $n_{(g)}$ design PSUs are placed into two groups called "variance PSUs" $s_{(g1)}$ and $s_{(g2)}$ such that no interviewer collects data in both $s_{(g1)}$ and $s_{(g2)}$; $\hat{Y}_{(g)} = \hat{Y}_{(g1)} + \hat{Y}_{(g2)}$ and $\hat{V}^*(\hat{Y}_{(g)}) = \left(\hat{Y}_{(g1)} - \hat{Y}_{(g2)}\right)^2$. Under the assumption that the expectations of $\hat{Y}_{(g1)}$ and $\hat{Y}_{(g2)}$ are equal and additional regularity conditions, $\hat{V}^*(\hat{Y})$ will be approximately unbiased or conservative for the combined variance of $\hat{Y}$. The remainder of this paper presents algorithms for construction of variance strata and PSUs that satisfy the condition that no interviewer collects data in both $s_{(g1)}$ and $s_{(g2)}$.

3. Algorithms for Optimal Pairing

We developed three algorithms to implement the general ideas of Section 2.

First, in some public-use datasets, variance strata and PSUs are not identified explicitly, but are defined implicitly by the inclusion or exclusion of sample elements in specific replicate-weight groups. Section 3.1 presents an algorithm for identification of current variance stratum and PSU membership based on the patterns of replicate weights assigned to each sample element.

Second, we need to re-group our PSUs into new variance strata and PSUs such that within a new stratum, data collected by a given interviewer will be contained in no more than one new variance PSU, which will in turn allow the computation of the new variance estimator (3). Section 3.2 presents a simple algorithm for this task.

Finally Section 3.3 presents an algorithm for assignment of replicate weights based on the newly-defined variance strata and PSUs, and Section 3.4 discusses the use of commercial software to implement the algorithms in Sections 3.1-3.3.

3.1 Algorithm for identification of current variance stratum and PSU membership from current replicate weight patterns

Before we can implement the ideas of Section 2, we need to identify the variance stratum and PSU assignments used in our current variance estimator (1), or replication-based versions thereof. In some cases, the public-use dataset and documentation provides explicit labels for these strata and PSUs. In other cases, the dataset omits this information and provides only the replicate weights needed for calculation of a variance estimator through balanced repeated replication. For the latter cases, note that sample elements have the same replicate weight patterns across the replicates if and only if they are in the same variance stratum. Also note that sample elements have complementary replicate weight patterns across the replicates if and only if they are in the same variance stratum but also in the other variance PSU. Therefore, we can use the replicate-weight pattern to identify variance stratum and variance PSU membership using the following algorithm:

1. Let Q be the number of replicate weights used for the current BRR procedure, and let $\tilde{w}_{jq}$ be the BRR weight for sample element j and replicate q. Define $w^*_{jq} = \{1 \, if \, \tilde{w}_{jq} \neq 0 \, ; \, 0 \, otherwise\}$. Change values of weights in such a way that new weights equal 0 if old weights were equal to 0; otherwise new weights equal 1. For cases in which the weights $\tilde{w}_{jq}$ were given by the Fay BRR method with a factor K, new weights equal 0 if old weights were equal to { full weights × K }otherwise new weights equal 1 (see Appendix A).

2. Order the data from Step 2 by values of new weights.

3. Group data from Step 3 by the same new weight pattern across the Q replicates. Each group of data is a variance PSU. Note that the first variance PSU and the last variance PSU are complementary and belong to the same variance stratum; the second variance PSU and the second to the last variance PSU are complementary and also belong to the same variance stratum. In the same manner, find all complementary variance PSUs, and identify variance stratum and variance PSU membership.

3.2 Algorithms for matching variance PSUs

Using the following algorithm, we assigned variance PSUs which do not share interviewers to each variance stratum:

1. Make a two-column dataset with variance PSU and interviewer variables.

2. Read the data from Step 1 into an array, b, so that b[i] is a set of interviewers in ith variance PSU. Note that a number of elements of b[i] may be different from a number of elements of b[j].

3. Let p be a number of variance PSUs. Create a $p \times p$ penalty matrix, M, with an arbitrary initial value in each cell.

4. Assign a value to M[i,j] according to the optimality criteria: for example, M[i,j]=0 if b[i] and b[j] are disjointed; M[i,j]=1 otherwise.

5. Find the row whose row sum is the largest. If k-th row has the largest sum, then k-th variance PSU is the hardest to match with other variance PSUs.

6. Let 'x' be the largest value of M[i,j]. Make the every element of k-th column larger than 'x', the largest number, so that k-th variance PSU will not be matched with other variance PSU later.

7. In the k-th row, find the index 'j' whose element is the smallest. Make the every element of j-th column larger than 'x', the largest number, so that j-th variance PSU will not be matched with other variance PSU later.

8. Match k-th variance PSU and j-th variance PSU. Add the matched pair [k,j] to the list of matched pairs, 'matched'.

9. Make every element of j-th row and k-th row 0 so that j-th row and k-row will not be chosen as the ones with the maximum row sum.

10. Repeat the procedure by choosing the row with next largest row sum.

Note that there is no unique pairing in this case. We can reward a pairing such as one with equal numbers of interviewers or any kind at Step 4.

3.3 Algorithms for attaching new BRR weights

Each rematched pair in a variance stratum did not share the common interviewers. Choose the number of replicates, Q, in such a way that Q is the next higher multiple of 4 of the number of variance strata, G. We also have the full-sample set of weights. For the purpose of the current BRR procedure, we treated the weights as fixed. Consequently, this BRR method will not account explicitly for the additional components of variability associated with other weight adjustment steps, and the dependence of these additional steps on the other units included in the sample. We assigned weights to rematched variance PSUs using the following algorithm:

1. From a $Q \times Q$ Hadamard matrix, choose only G variance strata with Q replicate weights.

2. Choose one variance PSU from each variance stratum based on the Hadamard matrix by choosing, say, the first group, if the corresponding entry value of the Hadamard matrix is 1.

3. Join the new weight (0 or 1) from Step 2 with a data with variance strata and PSU variables.

4. Merge new weight data from Step 3 with the original data by variance PSU variable.

5. Compute {new weight × final weights × 2} for new BRR weights.

3.4 Use of Commercial Software

The previous three subsections presented three algorithms. In implementation of the first two algorithms with commercial software, we found it important to use packages with the following characteristics.

First, to implement the membership-identification algorithm in Section 3.1, we needed a package that enables us to group sampling units with the same replicate weight patterns across the replicates. We found that SAS had the features required: SAS Proc Sort lets us sort data by multiple variables such as the Q replicate weights in our example. In choosing software, SAS, being the standard statistical software, at BLS was our first choice.

Second, to implement the variance PSU matching algorithm in Section 3.2, it was necessary to conduct all possible pairwise comparisons among variance PSUs, and see whether two variance PSUs share common interviewers. Note that a number of interviewers of a variance PSU may differ from the number of interviewers of the other variance PSU. In Maple, defining an unbalanced array is straightforward. In SAS however, we found it a challenging task. Maple has a rich set of built-in data structures such as sequences, lists, and sets (Heal et al. 1996). It is much smoother to implement a collapsing procedure using Maple, due to its specific features.

4. Applications to the U.S. Consumer Expenditure Survey

The CE uses two modes of data collection, diary and interview. The principal reason for this use of multiple collection modes is that some expenditures (generally small or frequently purchased items) are believed to be more readily captured through a diary, while other items (generally purchases that are larger, less frequent, or otherwise more salient) are more readily captured through a periodic in-person interview (Eltinge et al., 2000). Expenditures are reported at a relatively fine level of aggregation known as the six digit Universal Classification Code (UCC) level (Eltinge et al., 2000).

We considered only the components of the mean monthly expenditure of the CE Interview Survey that contribute to current CE production estimates. In particular, we exclude interview data collected for UCCs that are published on the basis of diary data only. Consequently, the “Overall Mean” entries are based on data from the 432 UCCs for which publication is based on the interview reports. In addition, the entries for “Apparel” are based on apparel UCCs that are published from interview data; similarly for home furnishings; travel and utilities.

The data used for this analysis was generated from the monthly expenditures (MTABQ) files and the CU characteristics and income (FMLYQ) files of the Phase 3 databases. For computing the mean monthly expenditure, we used the UCCs collected in the Interview Survey and used for publishing quarterly expenditures (see 2000 Interview Stub Parameter File). We didn’t include Pension and Social Security expenditures which were stored in the income (ITAB) file.

4.1 PSU Characteristics Used in the Stratum Collapse Method

The original sample design included 105 PSUs (31 self-representing and 74 non self-representing). Three self-representing PSUs were each randomly partitioned into four variance PSUs; for each of these three cases, the four variance PSUs were grouped into two variance strata. In addition, four self-representing PSUs were each randomly partitioned into two variance PSUs. In each of these four cases, the resulting pair of variance PSUs formed a variance stratum. Also, 24 self-representing PSUs were placed into 12 pairs. Each of these 24 PSUs was then randomly partitioned into two half-PSUs, and matched with a half-PSU from the other PSU in its pair. This resulted in another 24 variance PSUs. Six of the large non self-representing PSUs became variance PSUs, and were paired to form a total of three variance strata. Forty-four non self-representing PSUs were paired to form 22 variance PSUs contained in 11 variance strata. Finally, 24 PSUs were grouped into eight variance PSUs containing three original PSUs each; these in turn were grouped into four variance strata. In summary, all of the self-representing PSUs were split into two or four sections in forming the variance PSUs. None of the non self-representing PSUs were split. Every one of the 80 variance PSUs could be mapped back to 105 original PSUs.

Therefore, each non self-representing PSU is contained entirely within one variance PSU. Thus, for variance PSUs constructed from non self-representing PSUs, interviewers are contained in more than one variance PSU only if that interviewer was in more than one original PSU. However, each self-representing PSU is contained in either two or four variance PSUs. Consequently, for variance

PSUs constructed from self-representing PSUs, interviewers may be in more than one variance PSU solely because of the random partition process.

Note that self-representing PSUs were collapsed only with self-representing ones, and non self-representing PSUs were collapsed only with non self-representing ones. Hence we call a variance PSU formed with self-representing PSUs a self-representing variance PSU, and a variance PSU formed with non self-representing PSUs a non self-representing variance PSU.

The CE Interview Survey had 792 interviewers in Year 2000, and more than half of those interviewers collected data in more than one variance PSU.

5. Discussion

Depending on the algorithms, it is sometimes necessary to run a number of iterations to assign variance PSUs which do not share the interviewers to each variance stratum for all variance strata. We have currently developed an algorithm which shows the separability (No-Common-Interviewers) through one iteration. It is possible to incorporate various optimality criteria in the objective function matrix used to construct a given set of variance PSUs and strata. For example, one may match pairs on the basis of the number of interviewers assigned to a variance PSU, or on the basis of the number of interviewers that a given variance PSU shared with other variance PSUs, or on the basis of population characteristic variable in variance PSU pairing.

6. References

Bailar, B., Bailey, L., and Stevens, J. (1977). Measures of Interviewer Bias and Variance. *Journal of Marketing Research*, Vol. XIV, pp. 337-43.

Eltinge, J.L., Cho, M.J. and Lahiri, P. (2005). Variance Estimation and Inference from Complex Survey Data in the Presence of Interviewer-Level Measurement Error. Paper to be presented at the Federal Committee on Statistical Methodology (FCSM) Research Conference 2005.

Eltinge, J.L., Sukasih, A. and Weber, W. (2000). Feasibility of Constructing Combined Estimators using Consumer Expenditure Interview and Diary Data. Manuscript, Office of Survey Methods Research, U.S. Bureau of Labor Statistics.

Fay, R.E. (1989). Theory and Application of Replicate Weighting for Variance Calculations. *Proceedings of the American Statistical Association* , Section on Survey Research Methods.

Fuller, W.A. (1970). Sampling with Random Stratum Boundaries. *Journal of the Royal Statistical Society*, Series B, Methodological, 32, 209-226.

Greenwood, M. (1946). Proceedings of a meeting of the Royal Statistical Society held on July 16th, 1946. *Journal of the Royal Statistical Society*, Vol. 109, No. 4 (1946), pp. 325-378.

Hansen, M.H., Hurwitz, W.N. and Madow, W.G. (1953). *Sample Survey Methods and Theory.* Vol. I. Methods and Applications. Vol. II. Theory. John Wiley and Sons, New York.

Hartley, H.O., Rao, J.N.K. and Kiefer, G. (1969). Variance Estimation with One Unit per Stratum. *Journal of the American Statistical Association*, 64, pp. 117-123.

Heal, K.M., Hansen, M.L. and Rickard, K.M. (1996). *Maple V - Learning Guide.* Springer-Verlag, New York.

Judkins, D.R. (1990). Fay's Method for Variance Estimation. *Journal of Official Statistics*, Vol. 6. No. 3, pp. 223-239.

Kish, L. (1962). Studies of Interviewer Variance for Attitudinal Variables. *Journal of the American Statistical Association*, Vol.57, No. 297, 92-115.

Kish, L. (1995). *Survey Sampling.* John Wiley and Sons, New York.

Korn, E.L. and Graubard B.I. (2003). Estimating Variance Component by Using Survey Data. *Journal of the Royal Statistical Society*, B, 65, Part 1, pp 175-190.

O'Muircheartaigh, C. and Campanelli, P. (1999). A Multilevel Exploration of the Role of interviewers in Survey Non-Response. *Journal of the Royal Statistical Society*, Vol.162, No. 3, 437-446.

O'Muircheartaigh, C. and Campanelli, P. (1998). The Relative Impact of Interviewer Effects and Sample Design Effects on Survey Precision. *Journal of the Royal Statistical Society*, Vol.161, No. 1, 63-77.

Rust, K. and Kalton, G. (1987). Strategies for Collapsing Strata for Variance Estimation. *Journal of Official Statistics*, Vol. 3, No. 1, 1987, pp.69-81.

Skinner, C.J., Holt, D. and Smith, T.M.F., Eds (1989). *Analysis of Complex Surveys.* John Wiley and Sons, New York.

Wolter, K.M. (1985). *Introduction to Variance Estimation.* Springer-Verlag, New York.

7. Appendix A: Modifications of Section 3.1 for Fay BRR Weights

For some general background on the Fay BRR method, see Fay (1989) and Judkins (1990). Standard BRR methods define replicate weights $\tilde{w}_{jq}$ that (for a given replicate q and sample element j) are equal to either zero or approximately $2 \times w_{j\,Full}$, the full-sample weight assigned to element j in computation of $\hat{Y}$, say. In contrast with this standard method, the Fay method defines replicate weights $w_{jq\,Fay}$ that (under conditions) are approximately equal to $\{(2-K) \times w_{j\,Full}\}$ or $\{K \times w_{j\,Full}\}$ where K equals some value in the interval $(0, 1)$; For example, Fay (1989) considers use of $K = 0.5$. The above-mentioned approximation is not exact in cases in which one reproduces weight modification steps (e.g. poststratification for each set of replicate weights), as generally preferred under a theoretically rigorous development. In some special cases, these modifications can lead to substantial variability in the resulting replicate weights $w_{jq\,Fay}$.

With the exception of extreme cases of the above-mentioned variability, one may identify membership in current variance strata and PSUs from Fay BRR weights through the algorithm of Section 3.1, with step 2 replaced by $2'$ Define $w^*_{jq} = \{1\ if\ \frac{w_{jq\,Fay}}{w_{j\,Full}} > 0.5\,;\ 0\ otherwise\}$.

8. Appendix B

8.1 SAS codes for finding current replicate weight patterns

```
* Change values of weights to 1 or 0 values to find weight pattern easily;
* If old weight > 0 then new weight = 1, and old weight =0 then new weight = 0;
* In our example, we have 44 replicate weights;

data newdata (drop= I);
set olddata;
array Weight(44) wtrep01-wtrep44;
do I=1 to 44;
if Weight(I)>0 then Weight(I)=1;
end;
run;

* Sort data by replicate weights (1 or 0);

proc sort data= newdata;
by wtrep01-wtrep44;
run;

* Note that variance PSUs have the same weight patterns across the replicates;

* Assign new identification number to variance PSUs;

data final;
set newdata;
by wtrep01-wtrep44;
retain ID 0;
if first.wtrep44 then ID +1;
run;
```

8.2 Maple codes for matching of PSUs

```
♯ A '♯' character causes Maple to ignore all remaining text on the line, and is used for comments like this one.

♯ ♯ creating macros: maxi, mini, maxind, minind ♯ ♯
♯ an example 'a' is a list
> restart:
> a:=[1,4,5,2,3,3, 1, 3,5]:
♯ maxi is the maximum value of observations
> maxi:=a->sort(a)[nops(a)]:
♯ mini is the minimum value of observations
> mini:=a->sort(a)[1]:
♯ if value < max, then keep adding index
> maxind:=proc(a) local i,k;
> k:=1: for i from 1 to nops(a)
> while a[i]<maxi(a) do k:=k+1;
> end do; k;
> end:
♯ if value > min, then keep adding index
> minind:=proc(a) local i,k;
> k:=1: for i from 1 to nops(a)
> while a[i]>mini(a) do k:=k+1;
> end do; k;
> end:
♯ example of actual values of maxi, mini, maxind,
♯ minind for 'a'
> a; maxi(a); mini(a); maxind(a); minind(a);
[ 1, 4, 5, 2, 3, 3, 1, 3, 5 ]
5
1
3
1

♯ ♯ creating macro: rs which is row sum for each row ♯ ♯
♯using an example 'm'
> n:=4:
> m:=[seq([seq(i*j,i=1..n)],j=1..n)]:
> i:='i': j:='j':
> rs:=proc(m) local i,n, an: n:=nops(m[1]);
> for i to n do;
> an[i]:=sum(m[i,j],j=1..n): od;
> [seq(an[i],i=1..n)];
> end:

♯ ♯ numerical results using an example 'm'♯ ♯
♯ display 'm';
> m;
```

```
[ [1, 2, 3, 4], [2, 4, 6, 8], [3, 6, 9, 12], [4, 8, 12, 16]]
♯ display row sums for matrix ‘m’;
> rs(m);
[10, 20, 30, 40]
♯ display which row has the maximum row sum of ‘m’
> maxind(rs(m));
4
♯ display which row has the minimum row sum in ‘m’;
> minind(rs(m));
1
♯‘map’ is a built-in function is to apply ‘maxi’ function
♯ to each row of ‘m’; ‘m’ has four rows,
♯ and in each row of ‘m’, it will display maximum
> map(maxi,m);
[4, 8, 12, 16]

♯ ♯ creating macro ‘fm’ ♯ ♯
♯ fm (first match) is a key macro which uses all previous
♯ macros
♯ Copy the matrix ‘m’ to ‘mt’ so that ‘m’ is not changed.
♯ Find maximim value of each row of ‘mt’,
♯ and find the maximum value
♯ among those maximum values, and call it ‘x’.
♯ Put matched pairs to ‘matched’.
♯ ‘matched’ is empty at the beginning;
♯ Find the row whose rowsum is the largest.
♯ Make the every element of k-th column larger
♯ than ‘x’, the largest number so that k-th PSU
♯ will not be matched with other group later.
♯ In the k-th row,
♯ find the index ‘j’ whose element is the smallest.
♯ Make the every element of j-th column larger
♯ than the largest number, ‘x’ so that
♯ j-th PSU will not be matched with other group later.
♯ Match k-th group and j-th group.
♯ Add the matched pair [k,j] to
♯ the list of matched pairs, ‘matched’.
♯ Make every element of j-th row and k-th row 0
♯ so that j-th row and k-row will not be chosen
♯ as the ones with the maximum row sum.
♯ Repeat the procedure by choosing the row
♯ with maximum row sum.

> fm:=proc(m) local i,x,j,mt,k,ii,jj, c, matched;
> mt:=m: x:=maxi(map(maxi,mt)): matched:=[ ]:
for c from 1 to nops(m)/2
> do
> k:=maxind(rs(mt)):
> for jj from 1 to nops(m) do mt[jj,k]:=x+1: end do:
> j:=minind(mt[k]): for ii from 1 to nops(m) do
mt[ii,j]:=x+1: end do:
> matched:=[op(matched),[k,j]];
for jj from 1 to nops(m) do mt[k,jj]:=0:
> mt[j,jj]:=0:end do: end do: matched; end:
```

```
♯ ♯ In the example ‘m’, ♯ ♯
♯ the largest value in ‘m’, maxi(map(maxi,m))=16.
♯ hence x= 16.
♯ The fourth row has the largest row sum.
♯ Set every element in the fourth column to 17
♯ which is x+1.
♯ The first element of the fourth row
♯ has the smallest value.
♯ Set every element in the first column to 17
♯ which is x+1.
♯ Match the fourth group with the first group,
♯ and now ‘matched’ is [[4,1]].
♯ Then replace every element in the first and
♯ the fourth rows to 0. Repeat the procedure
♯ by choosing the row with maximum row sum.
♯ Note the third row has the largest row sum.
♯ Finally, ‘matched’ is [[4,1],[3,2]].
> m;
[ [1, 2, 3, 4], [2, 4, 6, 8], [3, 6, 9, 12], [4, 8, 12, 16]]
> fm(m);
[[4, 1], [3, 2]]
> n:=‘n’:

♯ ♯ Define ‘a’ the list of the data. ♯ ♯
♯ “c:/mydata.txt” has variance PSU
♯ and interviewers ID information.
> a:=readdata(“c:/mydata.txt”, integer,2):
♯ mm is the number of entries.
> mm:=nops(a);
mm := 1655
♯ p is the number of variance PSUs.
> p:=80:
♯ b is the list of each variance PSU, and
♯ elements of b[i] are interviewers ID numbers
♯ in ith variance PSU.
> b:=[seq({}, i=1..p)]:
> for i from 1 to p do
> for j from 1 to mm do
> if evalb(a[j,1]=i) then b[i]:={op(b[i]),a[j,2]}; end if;
> end do;
> end do;
> i:=‘i’: j:=‘j’:
♯ Example of the third variance PSU in our example.
> b[3];
{204, 205, 210, 215, 216, 221, 230, 240, 243, 247, 249,
250, 255, 259, 265, 267, 357, 358, 360, 383, 387, 391,
398, 400, 408}
♯ Construct a p × p penalty matrix where p=80
♯ in our example.
> m:=[seq([seq(2,i=1..p)],j=1..p)]:
> for i from 1 to p do
> for j from 1 to p do
> if evalb(b[i] intersect b[j] ={}) then m[i,j]:=0: else
> m[i,j]:=1:
> end if;
> end do;
> end do;
> i:='i': j:='j':
> result:=fm(m);
```

8.3 SAS codes for attaching new BRR weights

```
* The following SAS codes describe steps 1-3 in Section 3.3;
* In our example, we have 40 variance strata.
Hence, we chose 44*44 Hadarmard matrix;
* We changed '+' and '-' in the Hadarmard matrix
to 1 and 0, respectively, and saved it to C:\ MyDirectory \ 44Pal.xls;
* We then imported the worksheet from SAS;

proc import out=Hadarmard44 datafile= "C:\ MyDirectory \ 44Pal.xls" replace; getnames=No;
run;

* 'in.paired' is 80*2 matrix which has two variables, variance strata and variance PSU;
* Sort 'in.paired' by variance strata, and named output file 'paired';

libname in 'C:\ MyDirectory ';

proc sort data=in.paired out=paired;
by strata;
run;

proc iml;
use Hadarmard44;

* remove the first row of Hadarmard whose entries are 1 and then all 0 values;

Hadarmard = Hadarmard[2:44,];

RepWt = j(80,44,0);
do i=1 to 40;
do j=1 to 44;

* Choose a variance PSU from each variance stratum based on the Hadamard matrix;
* For example:;
* if Hadarmard[,]=1 then pick the first one by setting first one's value=1;
* if Hadarmard[,]=0 then pick the second one by setting second one's value=1;

if Hadarmard[i,j]=1 then RepWt[(2*i -1),j]=1;
else if Hadarmard[i,j]=0 then RepWt[(2*i),j]=1;
else RepWt[i,j]=0;
end;
end;

* RepWt (80*44) is a matrix of replicate weights, 1 or 0, based on Hadarmard matrix;
* Create NewWt by adding variance strata and variance PSUs information to RepWt

use paired;
NewWt = paired || RepWt;
name ={Strata PSU wtnew01 wtnew02 wtnew03
wtnew04 wtnew05 wtnew06 wtnew07 wtnew08 wtnew09
wtnew10 wtnew11 wtnew12 wtnew13 wtnew14 wtnew15
wtnew16 wtnew17 wtnew18 wtnew19 wtnew20 wtnew21
wtnew22 wtnew23 wtnew24 wtnew25 wtnew26 wtnew27
wtnew28 wtnew29 wtnew30 wtnew31 wtnew32 wtnew33
wtnew34 wtnew35 wtnew36 wtnew37 wtnew38 wtnew39
wtnew40 wtnew41 wtnew42 wtnew43 wtnew44};
create in.NewWt from NewWt [colname=name];
append from NewWt;
quit;
```