NIASRA
NATIONAL INSTITUTE FOR APPLIED
STATISTICS RESEARCH AUSTRALIA

National Institute for Applied Statistics Research Australia

The University of Wollongong

Working Paper

08-14

Model-Assisted Optimal Allocation for Planned Domains Using Composite Estimation

Wilford B. Molefe and Robert Graham Clark

National Institute for Applied Statistics Research Australia, University of Wollongong,
Wollongong NSW 2522. Phone +61 2 4221 5435, Fax +61 2 4221 4845. Email:
anica@uow.edu.au

MODEL-ASSISTED OPTIMAL ALLOCATION FOR PLANNED DOMAINS USING COMPOSITE ESTIMATION

Wilford B. Molefe [1] and Robert Graham Clark [2]

ABSTRACT

This paper develops allocation methods for stratified sample surveys where composite small area estimates are a priority, and area are used as strata. Longford(2006) proposed a objective criterion for this situation, based on a weighted combination of the mean squared errors of small area means and a grand mean. Here, we redefine this approach within a model-assisted framework, allowing regressor variables and a more natural interpretation of results using an intra-class correlation parameter. We also consider several uses of power allocation, and allow the placing of other constraints such as maximum relative root mean squared errors for stratum estimators. We find that a simple power allocation can perform very nearly as well as the optimal design even when the objective is to minimize Longford's (2006) criterion.

Key Words: small area estimation, sample design, sample size allocation, composite estimation, mean squared error.

[1] Department of Statistics, University of Botswana. E-mail: molefewb@mopipi.ub.bw

[2] National Institute for Applied Statistics Research Australia, University of Wollongong. Email: rclark@uow.edu.au

1. INTRODUCTION

Sample surveys have long been used as cost-effective means for data collection but it is also the case that general purpose surveys will often not achieve adequate precision for statistics for subpopulations of interest (often called domains or areas). Domains may be geographically based areas such as states. They may also be cross-classifications of a small geographic area and a specific demographic or social group. A domain is regarded as *small* if the domain-specific sample is not large enough to produce a direct estimate with reliable precision.

In this paper, we suppose that stratified sampling is used with H strata defined by the small areas, indexed by $h \in U^1$. In practice, small areas cannot always be identified in advance (e.g. Marker 2001), but in some cases they may, and even an imperfect identification of output areas of interest should result in an improved design. The population of units, indexed by j, is denoted U, of size N. The population of N_h units in stratum h is U_h and the sample of n_h units selected by simple random sampling without replacement (SRSWOR) from stratum h is s_h. Let Y_j be the value of the characteristic of interest for the jth unit in the population. The small area population mean for stratum h is $\bar{Y}_h$ and the national mean is $\bar{Y}$. The corresponding sample estimators are $\bar{y}_h$ and $\bar{y}$, respectively; $\bar{y}_h = n_h^{-1} \sum_{j \in s_h} y_j$ and $\bar{y} = \sum_{h \in U^1} P_h \bar{y}_h$, where $P_h = N_h/N$. Let the sampling variances be $v_h = var_p(\bar{y}_h)$ and $v = var_p(\bar{y})$.

Longford (2006) considers the problem of optimal sample sizes for small area estimation for this design. The approach is based on minimizing the weighted sum of the mean squared errors of the planned small area mean estimates and an overall estimate of the mean. The weight attached to each area is the area population raised to the q-th power, so the value of $0 \leq q \leq 2$ specifies the relative importance of larger compared to smaller areas. The mean squared error of the all-strata mean estimate is multiplied by G, where G reflects the perceived priority of this estimate. An analytical solution exists for the case where $G = 0$, but it has undesirable practical properties, and may sometimes result in zero or minimum sample sizes for some strata. When $G > 0$, Longford (2006) suggests numerical optimization.

Choudhry et al. (2012) investigate the use of nonlinear programming (NLP) to efficiently allocate sample to strata, when there may bounds on stratum sample sizes, and priority on overall, stratum and cross-strata domain estimates of multiple variables. The paper mostly concentrates on design-based direct area estimates, but they also consider the objective criterion of Longford (2006) for composite estimation. For the Canadian Monthly Retail Trade Survey, they show that the Longford allocation gives extremely unequal sample sizes by strata, for q equal to 0.5, 1 and 1.5. For example, when $q = 1.5$, the highest stratum coefficient of variation (CV) is 112%, and even for $q = 0.5$, the highest coefficient of variation is 24%, which was deemed too high. It is not clear whether these CV%s refer to direct or composite estimators - such high CVs would be surprising for composite estimators, as their CVs are bounded above even as the sample size tends to zero. Choudhry et al. (2012) did not investigate whether other designs such as power allocations can give low values of Longford's criteria.

The aim of this paper is to find the best allocation to strata for a linear combination of the mean squared errors of small area composite estimates and of an overall estimator of the mean, similar to Longford (2006). In Section 2 we reformulate the objective in model-assisted terms, introduce the use of regressor variables, and derive a model-assisted composite estimator. Section 3 is devoted to optimizing the design. In Subsection 3.1 we discuss direct optimization, for example by NLP. Subsection 3.2 describes power allocation with the exponent chosen to numerically minimize the objective criterion. Section 4 is a numerical study of the various methods using the Swiss canton data of Longford (2006) and Section 5 contains conclusions.

2. COMPOSITE ESTIMATION

Composite estimators for small areas are defined as convex combinations of direct (unbiased) and synthetic (biased) estimators. A simple example is the composition $(1-\phi_h)\bar{y}_h + \phi_h \bar{y}$ of the sample mean $\bar{y}_h$ for the target area h and the overall sample mean $\bar{y}$ of the target variable. The coefficients ϕ_h are set with the intent to minimise its mean squared error (MSE), see for example Rao (2003, section 4.3). The coefficients by which the MSE is minimized depend on some unknown parameters which have to be estimated.

Better results can be obtained if there are some regressors $\boldsymbol{x_i}$, for which domain population means are available, as well as sample data at either unit or domain level enabling Y to be regressed on $\boldsymbol{x}$. A synthetic estimator for domain h is then defined by $\hat{\bar{Y}}_{h(syn)} = \hat{\boldsymbol{\beta}}^{\boldsymbol{T}} \bar{\boldsymbol{X}}_{\boldsymbol{h}}$, where $\hat{\boldsymbol{\beta}}$ is the estimated regression coefficient, and $\bar{\boldsymbol{X}}_{\boldsymbol{h}}$ is the domain population mean of the regressor variables. An efficient direct estimator which is particularly suitable when domain sizes may be small is $\bar{y}_{hr} = \bar{y}_h + \hat{\boldsymbol{\beta}}^{\boldsymbol{T}} \left(\bar{\boldsymbol{x}}_{\boldsymbol{h}} - \bar{\boldsymbol{X}}_{\boldsymbol{h}} \right)$ (Hidiroglou and Patak, 2004) where $\bar{y}_h$ and $\bar{\boldsymbol{x}}_{\boldsymbol{h}}$ are the domain h sample means of Y and X. A composite estimator can then be constructed as $\tilde{y}_h^{\mathcal{C}} = (1-\phi_h)\bar{y}_{hr} + \phi_h \hat{\boldsymbol{\beta}}^{\boldsymbol{T}} \bar{\boldsymbol{X}}_{\boldsymbol{h}}$.

The design-based MSE of the composite estimator is given by:

$$MSE_p\big(\tilde{y}_h^{\mathcal{C}};\bar{Y}_h\big) = (1-\phi_h)^2 v_{hr} + \phi_h^2\{v_{h(syn)} + B_h^2\} + 2\phi_h(1-\phi_h)c_h$$

where c_h is the sampling covariance of $\bar{y}_{hr}$ and $\hat{\bar{Y}}_{h(syn)}$, v_{hr} is the sampling variance of the direct estimator $\bar{y}_{hr}$, $v_{h(syn)}$ is the sampling variance of the synthetic estimator $\hat{\bar{Y}}_{h(syn)}$, and $B_h = \boldsymbol{\beta_U^T}\boldsymbol{\bar{X}_h} - \bar{Y}_h$ is the bias of using $\hat{\bar{Y}}_{h(syn)}$ to estimate $\bar{Y}_h$, with $\boldsymbol{\beta_U}$ denoting the approximate design-based expectation of $\boldsymbol{\hat{\beta}}$. Further,

$$MSE_p\big(\tilde{y}_h^{\mathcal{C}};\bar{Y}_h\big) \approx (1-\phi_h)^2 v_{h(syn)} + \phi_h^2 B_h^2 \tag{1}$$

because $c_h \ll v_h$ and $v \ll v_h$ when number of small areas is large, under regularity conditions.

A two-level linear model ξ conditional on the values of $\boldsymbol{x}$ will be assumed, with uncorrelated stratum random effects u_h and unit residuals ϵ_i:

$$\left.\begin{aligned}
&Y_i = \boldsymbol{\beta^T}\boldsymbol{x_i} + u_h + \epsilon_i\\
&E_\xi\left[u_h\right] = E_\xi\left[\epsilon_i\right] = 0\\
&var_\xi\left[u_h\right] = \sigma_u^2\\
&var_\xi\left[\epsilon_j\right] = \sigma_e^2
\end{aligned}\right\} \tag{2}$$

for $h \in U^1$ and $i \in U_h$. This implies that $var_\xi\left[Y_i\right] = \sigma_u^2 + \sigma_e^2 = \sigma^2$ for all $i \in U$, and that the covariance $cov_\xi\left[Y_i, Y_j\right]$ equals $\rho\sigma^2$ for units $i \neq j$ in the same strata and 0 for units from different strata, where $\rho = \sigma_u^2/\left(\sigma_u^2 + \sigma_e^2\right)$.

Under the model (2),

$$E_\xi[v_{hr}] = E_\xi\big[n_h^{-1}S_{hw}^2\big] = n_h^{-1}\sigma_h^2(1-\rho)$$

where S_{hw}^2 is the within-stratum-h sample variance of $y_i - \boldsymbol{\beta_U^T}\boldsymbol{x_i}$; and

$$\begin{aligned}
E_\xi[B_h^2] &= E_\xi\big[\big(\bar{Y}_h - \bar{Y}_{h(syn)}\big)^2\big] \approx E_\xi\big[\big(\bar{Y}_h - \boldsymbol{\beta^T}\boldsymbol{\bar{X}_h}\big)^2\big]\\
&= var_\xi[\bar{Y}_h] = \sigma_h^2 N_h^{-1}[1 + (N_h - 1)\rho].
\end{aligned}$$

To simplify expressions, we assume that n, N_h and H are all large, although we do not derive rigorous asymptotic results. Assuming that N_h is large, we firstly obtain $E_\xi[B_h^2] \approx \sigma_h^2\rho$. Substituting for $E_\xi[v_{hr}]$ and $E_\xi[B_h^2]$ into (1) we get the anticipated MSE or approximate model assisted mean squared error, denoted $AMSE_h$:

$$AMSE_h = E_\xi MSE_p\big(\tilde{y}_h^{\mathcal{C}}; \bar{Y}_h\big) \approx (1-\phi_h)^2 n_h^{-1}\sigma_h^2(1-\rho) + \phi_h^2\sigma_h^2\rho \tag{3}$$

Optimizing with respect to ϕ_h we immediately obtain the optimal weight ϕ_h as:

$$\phi_{h(opt)} = (1-\rho)\big[1+(n_h-1)\rho\big]^{-1}. \tag{4}$$

We substitute the optimum weight (4) into (3) to obtain the approximate optimum anticipated MSE:

$$\begin{aligned} AMSE_h &= E_\xi MSE_p\big(\tilde{y}_h^{\mathcal{C}}\big[\phi_{h(opt)}\big]; \bar{Y}_h\big) \\ &\approx \Big(n_h\rho\big[1+(n_h-1)\rho\big]^{-1}\Big)^2 n_h^{-1}\sigma^2(1-\rho) + \Big((1-\rho)\big[1+(n_h-1)\rho\big]^{-1}\Big)^2\sigma^2\rho \\ &= \sigma_h^2\rho(1-\rho)\big[1+(n_h-1)\rho\big]^{-1}. \end{aligned}$$

3. OPTIMIZING THE DESIGN

3.1 Optimal Design for F

One way of measuring the performance of designs for small area estimation is with a linear combination of the anticipated MSE's of the small area mean and overall mean estimates. Following Longford (2006), but using anticipated MSEs instead of design-based MSEs, we define the criterion

$$\begin{aligned} F &= \sum_{h\in U^1} N_h^q AMSE_h + GN_+^{(q)} E_\xi var_p\left[\hat{\bar{Y}}_r\right] \\ &= \sum_{h\in U^1} N_h^q AMSE_h + GN_+^{(q)} E_\xi var_p\left[\sum_{h\in U^1} P_h\bar{y}_{hr}\right] \\ &\approx \sum_{h\in U^1} N_h^q AMSE_h + GN_+^{(q)} E_\xi \sum_{h\in U^1} P_h^2 n_h^{-1} S_{hw}^2 \\ &= \sum_{h\in U^1} N_h^q \sigma_h^2\rho(1-\rho)\big[1+(n_h-1)\rho\big]^{-1} + GN_+^{(q)} \sum_{h\in U^1} \sigma_h^2 P_h^2 n_h^{-1}(1-\rho) \end{aligned}$$

where the weights N_h^q reflect the inferential priorities for area h, with $0 \leq q \leq 2$, and $N_+^{(q)} = \sum_{h \in U^1} N_h^q$, and $\bar{y}_{hr}$ is the grand mean estimator defined in Section 2. This objective reflects the fact that surveys have many stakeholders, some of whom will be only concerned with one specific small area, while others will place priority only on national estimates. Estimates for small regions are often considered a priority, particularly if they correspond to administrative or governmental jurisdictions, although smaller areas may be assigned less priority than larger regions. The quantity G is a relative priority coefficient. Ignoring the goal of national estimation corresponds to $G = 0$ and ignoring the goal of small area estimation corresponds to large values of G, since when G is very large the second component in (5) dominates. The factor $N_+^{(q)}$ is introduced to appropriately scale for the effect of the absolute sizes of N_h^q and the number of areas on the relative priority G. The criterion in (5) is algebraically similar to the criterion in Longford (2006). Here, however, we adopt the model-assisted approach which treats the design-based inference as the real goal of survey sampling, but employs models to choose between valid randomization-based alternatives (e.g. chapter 6 of Särndal et al. 1992).

The minimization is subject to a fixed sample size constraint. It would be straightforward to extend this to a fixed cost constraint with different cost coefficients in different strata. It would also be straightforward to allow σ^2 or ρ to vary by strata.

When national estimation has no priority ($G = 0$), the solution for the number of units to be sampled from each strata is found by optimizing (5) subject to a fixed total sampling cost function. The unique stationary point for this optimization is

$$n_{h,opt.} = \frac{n\sigma_h \sqrt{N_h^q}}{\sum_{h \in U^1} \sigma_h \sqrt{N_h^q}} + \frac{1-\rho}{\rho}\left(\frac{\sqrt{\sigma_h N_h^q}}{H^{-1}\sum_{h \in U^1} \sqrt{\sigma_h N_h^q}} - 1\right) \qquad (5)$$

If there are other active constraints (e.g. minimum stratum sample sizes), or if $G > 0$, then (5) does not apply and F must be minimized numerically, for example by NLP as in Choudhry et al. (2012).

In practice it would almost always be appropriate to set $0 \leq q \leq 2$, with $q = 0$ corresponding to all areas being equally important regardless of size, and $q = 2$ giving much greater weight to larger areas. (The value of $q = 2$ would lead to proportional allocation if direct estimates were used rather than composite - see for example Bankier 1988.) In many cases $q = 1$ would be a sensible compromise. For example, this has been used to motivate power allocations (Bankier, 1988) for master household samples in Vietnam and South Africa (Kalton et al., 2005, para 76, page 89).

The first term in (5) is the optimal allocation for the direct estimator and corresponds to power allocation (Bankier, 1988). The second term will be positive for more populous areas (large N_h) and negative for less populous areas. Therefore, the allocation optimal for model-assisted composite estimation has more dispersed subsample sizes $n_{h,opt.}$ than the allocation that is optimal for direct estimators.

To understand the properties of the optimal allocation when $G > 0$, and to provide a non-iterative method, Molefe (2011, chapter 3) derived Taylor Series approximations to the optimal n_h, based on small ρ. Unfortunately, the problem turns out not to be amenable to Taylor series approximation, and the resulting allocations performed poorly in numerical studies and did not provide useful insights, and so are not further discussed here.

3.2 Power Allocation

Power allocations (Bankier, 1988) are defined by

$$n_h = \frac{nN_h^p}{\sum_{h \in U^1} N_h^p} \tag{6}$$

for $h \in U^1$, where $0 \leq p \leq 1$. A special case is the square root allocation when $p = \frac{1}{2}$. The exponent p is called the power of the allocation. Setting $p = 1$ results in proportional allocation and $p = 0$ results in equal allocation.

Bankier (1988) proposed choosing p based on perceived relative priorities. However, this was based on direct estimates being used in each stratum. We are interested in the case where composite estimation is to be used, and the objective is to obtain a low value for F in (5). We obtain numerically the value of p which minimizes F by one-dimensional optimization. We further consider imposing minimum stratum sample sizes, with p re-optimized accordingly.

4. NUMERICAL STUDY

We use data on the 26 cantons of Switzerland (Longford, 2006); their population sizes range from 15,000 (Appenzell-Innerrhoden) to 1.23 million (Zürich). The population of Switzerland is 7.26 million. We assume that $n = 10,000$, $\rho = 0.025$ and $\sigma/\mu = 1$. The last assumption only affects the magnitude of F and RRMSEs but not the relativities across methods. It is satisfied if, for example, a prevalence of 50% is estimated. All calculations were performed in the R statistical environment (R Development Core Team, 2012). Values of $q = 0, 0.5, 1, 1.5$ and 2, and values of $G = 0, 10$ and 100 were used, as in section 5.2 of Choudhry et al. (2012). The program used to produce all results is available in the appendix.

6 different allocations are evaluated in Tables 1-3. The value of F is shown for each design, relative to the value for equal allocation. Strata sample sizes were

constrained in all allocations to lie between 1 and the population sizes, while still summing to n. The first design is equal allocation, then proportional allocation. The third design is the optimal design, which minimizes F in (5) by NLP subject to all stratum sample sizes being at least 1. The fourth design minimizes F subject to all stratum relative root mean squared errors (RRMSEs) being 8% or less, which, from formula (5), is equivalent to a minimum stratum sample size of 113. For the third and fourth designs, NLP was carried out using the R package *Rsolnp* (Ghalanos and Theussl, 2011). The fifth design is power allocation, where the exponent p is calculated to minimize F. The sixth design is power allocation with all stratum sample sizes constrained to be 113 or more, and with p calculated to minimize F reflecting these constraints. In both cases, p was calculated using the *optimize* function in R.

Table 1 shows the efficiency of the various methods when $G = 0$, where efficiency refers to the achieved values of F from formula (5), which is a weighted combination of MSEs of area composite estimators and an overall grand mean estimator. When $q = 0$, equal allocation is then optimal for F, and all of the allocation methods except proportional allocation return equal allocation. For larger values of q, Optimal for Composite is the most efficient, as expected. Imposing the area maximum RRMSE constraint of 8% increases F by 4% when $q = 2$, and has negligible effect (1.4% or less) for smaller q. The optimal power allocation has virtually identical efficiency to the optimal-for-composite allocation, both with and without the area RRMSE constraint. The unconstrained optimal-for-composite and power allocations are more efficient than proportional allocation when q is small, and about equally efficient for $q \geq 1.5$. When the area RRMSE constraint is imposed, these designs suffer a small penalty, but are still more efficient than proportional except when $q = 2$.

Table 2 shows relative efficiencies for $G = 10$. As for when $G = 0$, the optimal-for-composite and optimal power designs perform very similarly, with a similar effect of imposing the area RRMSE constraint. The major difference compared to $G = 0$ is that proportional allocation is more efficient when G is larger. The optimal designs, even with the constraint imposed, remain more efficient than proportional allocation except for $q \geq 1.5$.

Table 3 shows efficiencies for large G (100). Here, proportional allocation is close to the best design for all q. It is about equivalent to the unconstrained optimal designs for all $q \geq 0.5$, and more efficient than the constrained optimal designs for all $q \geq 1$. The relative performance of the four optimal designs is about the same as for $G = 0$ and $G = 10$.

Figure 1 shows the distribution of the area RRMSEs across the 26 cantons for $q \in \{0.5, 1, 1.5, 2\}$ when $G = 0$ for the 4 optimal designs. The results for $q = 0$ are not shown because the canton sample sizes are then all equal for the optimal designs. The optimal for composite allocation (top left) shows a fairly tight range of area RRMSEs when $q = 0.5$, becoming more dispersed as q increases. The maximum RRMSEs are 6.6%, 9.4%, 13.8% and 15.6% for $q = 0.5, 1, 1.5$ and 2, respectively. Thus, for $q \geq 1$, some of the RRMSEs are undesirably large. The optimal for composite allocation with constraints forces all area RRMSEs to be 8% or less, shown by the top right panel. The bottom two panels show the corresponding optimal power allocations. The unconstrained power allocation is broadly similar to the unconstrained optimal for composite allocation, but less dispersed, with lower maximum area RRMSEs. The two constrained designs are very similar.

Table 4 shows the values of the optimal exponents calculated for the optimal power designs for each q and G. When G is 0 or 10, the optimal exponent p of the power allocation is very close to $q/2$, where q is the exponent in the definition of F in (5). For $G = 100$, the optimal exponent is quite close to 1, reflecting that for large G, F essentially reflects the variance of the grand mean, so that proportional allocation is nearly optimal. Table 5 shows the optimal power exponents when the area RRMSE constraints are applied. Applying these constraints has little effect on the optimal p.

Table 1: Relative efficiency of stratified designs for $G = 0$

Design	q=0	q=0.5	q=1	q=1.5	q=2
Equal allocation	1.000	1.000	1.000	1.000	1.000
Proportional allocation	2.117	1.340	0.887	0.637	0.493
Optimal for composite	1.000	0.933	0.786	0.627	0.488
Optimal for composite with constraints	1.000	0.933	0.787	0.636	0.509
Optimal power allocation	1.000	0.933	0.786	0.628	0.490
Optimal power with constraints	1.000	0.933	0.787	0.636	0.509

Table 2: Relative efficiency of stratified designs for $G = 10$

Design	q=0	q=0.5	q=1	q=1.5	q=2
Equal allocation	1.000	1.000	1.000	1.000	1.000
Proportional allocation	1.360	0.944	0.701	0.568	0.491
Optimal for composite	0.875	0.784	0.668	0.565	0.490
Optimal for composite with constraints	0.875	0.784	0.670	0.575	0.505
Optimal power allocation	0.905	0.791	0.668	0.565	0.490
Optimal power with constraints	0.905	0.790	0.670	0.575	0.505

Table 3: Relative efficiency of stratified designs for $G = 100$

Design	q=0	q=0.5	q=1	q=1.5	q=2
Equal allocation	1.000	1.000	1.000	1.000	1.000
Proportional allocation	0.656	0.576	0.529	0.503	0.488
Optimal for composite	0.608	0.565	0.527	0.503	0.488
Optimal for composite with constraints	0.608	0.567	0.536	0.515	0.501
Optimal power allocation	0.624	0.567	0.528	0.503	0.488
Optimal power with constraints	0.612	0.568	0.536	0.515	0.501

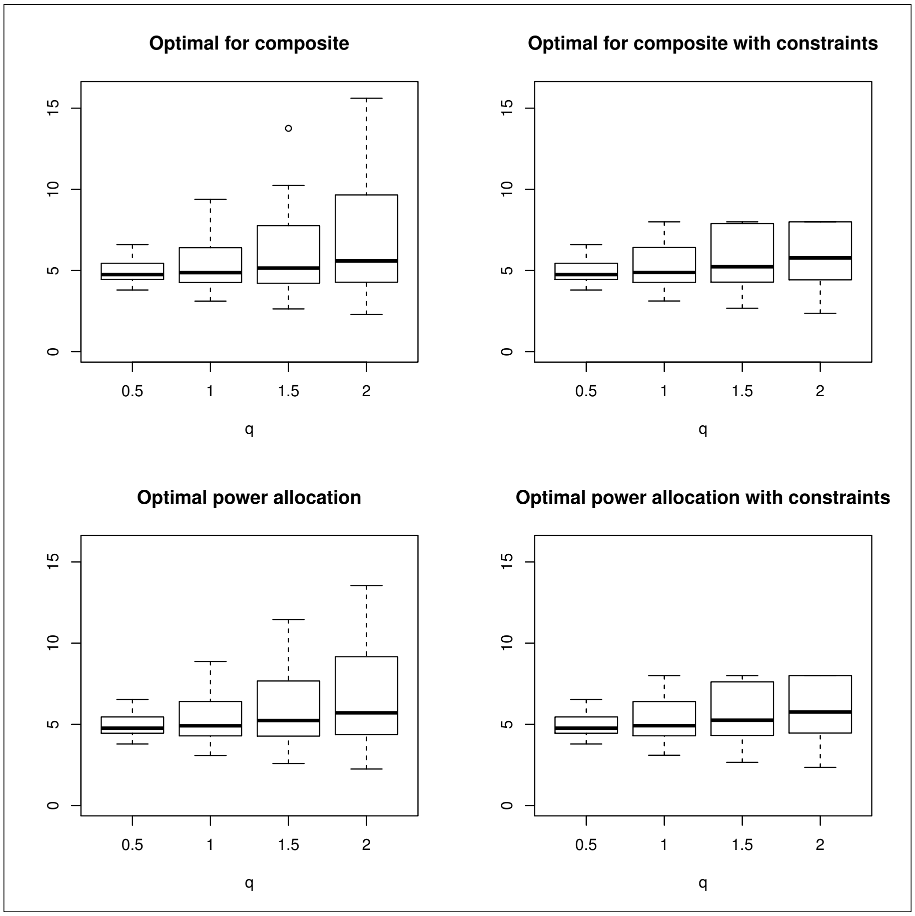

Figure 1: Distribution of anticipated relative root mean squared errors (RRMSE) (%) of estimated strata means for 4 allocations for various q with $G = 0$

Table 4: Optimal exponent in power allocation by G and q

	q=0	q=0.5	q=1	q=1.5	q=2
G=0	0.000	0.277	0.557	0.837	1.111
G=10	0.293	0.500	0.721	0.912	1.050
G=100	0.730	0.852	0.936	0.983	1.008

Table 5: Optimal exponent in power allocation by G and q with constraint on strata RRMSEs

	q=0	q=0.5	q=1	q=1.5	q=2
G=0	0.000	0.277	0.554	0.813	1.073
G=10	0.293	0.511	0.729	0.898	1.036
G=100	0.859	0.907	0.945	0.979	1.007

5. CONCLUSIONS

The anticipated MSE is a sensible objective criterion for sample design, because the particular sample which will be selected is not available in advance of the survey. Hence a criterion which averages over all possible samples is appropriate. Särndal et al. (1992, Chapter 12) base their optimal designs on the anticipated variance, which similarly averages over both model realizations and sample selection, although they consider only approximately design-unbiased estimators.

When both strata composite estimates and overall estimates are a priority, it makes sense to optimise an objective criterion which is a linear combination of the relevant anticipated MSEs. Allocations which are optimal in this sense give lower values of the objective function than either proportional or equal allocation. An optimal power allocation, $n_h \propto N_h^p$ where p is obtained numerically to minimize the objective function, is simpler and avoids the possibility of negative sample sizes which need to be truncated. It is also very nearly as efficient as the optimal allocation. When there is no priority on national estimation ($G = 0$), the optimal exponent turns out to be close to $p = q/2$, where q is the exponent applied to stratum population sizes in the objective criterion. This removes the need to perform an optimization. Thus, we recommend an objective criterion very similar to that of Longford (2006), but we suggest a simple power allocation with $p = q/2$ when $G = 0$, rather than the optimal allocation for F. This extends the the domain of application of power allocation to surveys using stratum composite estimators.

Rather than just relying on the overall objective criterion to appropriately balance resources across strata, it may often be desirable to also impose minimum stratum sample sizes or maximum stratum RRMSEs. These were successfully implemented using NLP. In the Swiss canton example in Section 4, an upper limit of 8% for stratum RRMSEs significantly reduced the highest RRMSE with little loss in the objective criterion. More complex constraints, for example on cross-strata domains or for multiple variables of interest, could also be implemented using NLP as described in Choudhry et al. (2012).

ACKNOWLEDGEMENTS: The authors wish to thank Professors Raymond Chambers and David Steel for their helpful suggestions on this paper.

REFERENCES

Bankier, M. D. (1988), “Power Allocations: Determining Sample Sizes for Subnational Areas,” *The American Statistician*, 42, 174–177.

Choudhry, G. H., Rao, J., and Hidiroglou, M. A. (2012), “On sample allocation for efficient domain estimation,” *Survey Methodology*, 38, 23–29.

Ghalanos, A. and Theussl, S. (2011), *Rsolnp: General Non-linear Optimization Using Augmented Lagrange Multiplier Method*, r package version 1.11.

Hidiroglou, M. A. and Patak, Z. (2004), “Domain Estimation Using Linear Regression,” *Survey Methodology*, 30, 67–78.

Kalton, G., Brick, J., and Lê, T. (2005), *Household Sample Surveys in Developing and Transition Countries*, United Nations: Statistics Division, Department of Economic and Social Affairs, no. 96 in Series F, `http://unstats.un.org/unsd/hhsurveys/pdf/Household_surveys.pdf` [accessed 24-May-2013].

Longford, N. T. (2006), “Sample Size Calculation for Small-Area Estimation,” *Survey Methodology*, 32, 87–96.

Marker, D. A. (2001), "Producing Small-Area Estimates from National Surveys: Methods of Minimizing use of Indirect Estimators," *Survey Methodology*, 27, 183–188.

Molefe, W. B. (2011), "Sample Design for Small Area Estimation," Ph.D. thesis, University of Wollongong, http://ro.uow.edu.au/theses/3495.

R Development Core Team (2012), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Rao, J. N. K. (2003), *Small Area Estimation*, Wiley.

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

APPENDIX: R PROGRAM

```
library(Rsolnp)

composite.objective <- function( nh , par.vector ){
  attach(par.vector)
  N <- sum(Nh)
  if(G==0) F <- sum( Nh^q * sigmasq.h * rho.h * (1-rho.h) / (1+(nh-1)*rho.h) )
  if(G==Inf) F <- sum( Nh^2 / N^2 / nh * sigmasq.h (1-rho.h) )
  if((G>0)&(G<Inf)) F <- sum( Nh^q * sigmasq.h * rho.h * (1-rho.h) / (1+(nh-1)*rho.h) ) +
                         G * sum(Nh^q) * sum( Nh^2 / N^2 / nh * sigmasq.h * (1-rho.h) )
  detach(par.vector)
  F
}

delta.fn.constrained.power <- function(lambda,par.vector,p,minsamp){
  nh <- pmax( par.vector$n * par.vector$Nh^p / sum(par.vector$Nh^p) * lambda , minsamp )
  sum(nh) - par.vector$n
}

obj.fn.p <- function( p , par.vector )
    composite.objective( nh=(par.vector$Nh)^p/sum((par.vector$Nh)^p)*par.vector$n ,
                                      par.vector=par.vector )

obj.fn.p.constrained <- function( p , par.vector , minsamp ){
  lambda <- uniroot( f=delta.fn.constrained.power , interval=c(0.1,1.01) , par.vector=par.vector ,
                                p=p , minsamp=minsamp)$root
  nh <- pmax( par.vector$n * par.vector$Nh^p / sum(par.vector$Nh^p) * lambda , minsamp )
  composite.objective( nh=nh , par.vector=par.vector)
}

all.allocations <- function( par.vector , method,stop=F){
  # par.vector should be a list with elements sigmasq.h, rho.h, Nh , minsamp, G , q, n, minCV
  # method = 1 (equal), 2 (proportional), 3=Longford-NLP,
  #     4=Longford-NLP with stratum-level CV constraints, 5=opt.power,
  #     6=constrained opt.power,  7=crude power, 8=constrained crude power
  if(stop) browser()
  N <- sum(par.vector$Nh)
  H <- length(par.vector$Nh)
  Ph <- par.vector$Nh / N
  nh.table <- matrix( data=NA , ncol=8 , nrow=H)
  nh.table[,1] <- rep(par.vector$n/H,H)
  nh.table[,2] <- par.vector$Nh/N*par.vector$n
  F.table <- rep( data=NA , 8 )
  totsamp.eqconstraints.fn <- function(nh,par.vector)   sum(nh)-par.vector$n
  method3.results <- solnp( pars = par.vector$Nh^(par.vector$q/2) /
                            sum(par.vector$Nh^(par.vector$q/2))*par.vector$n ,
                            fun=composite.objective , control=list(tol=1e-14),
                            eqfun=totsamp.eqconstraints.fn , eqB=0,
                            LB=par.vector$minsamp , par.vector=par.vector)
  nh.table[,3] <- method3.results$par
  # Longford-NLP with stratum CV constraint applied
  minsamp.cv <- (1-par.vector$rho.h) * ( 1/(par.vector$minCV/100)^2 - 1/par.vector$rho.h )
  minsamp.combined <- pmax( par.vector$minsamp , minsamp.cv )
  method4.results <- solnp( pars=nh.table[,1] ,
                            fun=composite.objective , control=list(tol=1e-14),
                            eqfun=totsamp.eqconstraints.fn , eqB=0, LB=minsamp.combined ,
                            par.vector=par.vector)
  nh.table[,4] <- method4.results$par
  # Optimal Power Allocation
  p.opt <- optimize( f=obj.fn.p , interval=c(0,2) , par.vector=par.vector )$minimum
  nh.table[,5] <- par.vector$Nh^p.opt / sum(par.vector$Nh^p.opt) * par.vector$n
  # Constrained Optimal Power Allocation
  p.opt.con <- optimize( f=obj.fn.p.constrained , interval=c(0,2) , par.vector=par.vector ,
                                      minsamp=minsamp.combined )$minimum
  lambda <- uniroot( f=delta.fn.constrained.power , interval=c(0.1,1.01) , par.vector=par.vector ,
                                p=p.opt.con , minsamp=minsamp.combined)$root
  nh.table[,6] <- pmax( par.vector$n * (par.vector$Nh/N)^p.opt.con /
                                    sum((par.vector$Nh/N)^p.opt.con) * lambda ,
                                    minsamp.combined )
  # Crude Power Allocation
  nh.table[,7] <- par.vector$n * (par.vector$Nh/N)^(par.vector$q/2) /
                          sum((par.vector$Nh/N)^(par.vector$q/2)) * lambda
  # Constrained Crude Power Allocation
  lambda <- uniroot( f=delta.fn.constrained.power , interval=c(0.1,1.01) ,
                                par.vector=par.vector , p=par.vector$q/2 ,
                                minsamp=minsamp.combined)$root
  nh.table[,8] <- pmax( par.vector$n * (par.vector$Nh/N)^(par.vector$q/2) /
                                    sum((par.vector$Nh/N)^(par.vector$q/2)) * lambda ,
                                    minsamp.combined )
  # return results
  for(method in c(1:8))
     F.table[method] <- composite.objective( nh=nh.table[,method] , par.vector=par.vector )
  list( nh.table=nh.table , F.table=F.table / F.table[1] ,
        optimal.power=p.opt , optimal.power.constrained=p.opt.con ,
        method3.results=method3.results , method4.results=method4.results)
}


#################################SWITZERLAND CANTON DATA (population sizes in thousands)
CantP <- c(551,53,15,261,187,947,239,414,38,186,69,351,167,39,33,453,73,131,
                  246,228,312,35,278,624,101,1229)*1e3
names(CantP) <- c("AG","AR","AI","BL","BS","BE","FR","GE","GL","GR","JU","LU","NE",
             "NW","OW","SG","SH","SZ","SO","TG","TI","UR","VS","VD","ZG","ZR")

# par.vector should be a list with elements sigmasq.h, rho.h, Nh , minsamp, G , q, n
```

```
allocs.G0.q1 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                         minsamp=rep(1,length(CantP)), G=0 , q=1, n=10000,minCV=8,
                         constraint=NULL) , stop=F)
allocs.G10.q1 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=10 , q=1, n=10000,minCV=8,
                         constraint=NULL) , stop=F)
allocs.G100.q1 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                             minsamp=rep(1,length(CantP)), G=100 , q=1, n=10000,minCV=8,
                         constraint=NULL) , stop=F)

allocs.G0.q2 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                         minsamp=rep(1,length(CantP)), G=0 , q=2, n=10000,minCV=8,
                         constraint=NULL) , stop=F)
allocs.G10.q2 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=10 , q=2, n=10000,minCV=8,
                         constraint=NULL) , stop=F)
allocs.G100.q2 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=100 , q=2, n=10000,minCV=8,
                         constraint=NULL) , stop=F)

allocs.G0.q0 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=0 , q=0, n=10000,minCV=8,
                         constraint=NULL) , stop=F)
allocs.G10.q0 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=10 , q=0, n=10000,minCV=8,
                         constraint=NULL) , stop=F)
allocs.G100.q0 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=100 , q=0, n=10000,minCV=8,
                         constraint=NULL) , stop=F)

allocs.G0.q0.5 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=0 , q=0.5, n=10000,minCV=8,
                         constraint=NULL) , stop=F)
allocs.G10.q0.5 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=10 , q=0.5, n=10000,minCV=8,
                         constraint=NULL) , stop=F)
```

```
allocs.G100.q0.5 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=100 , q=0.5, n=10000,minCV=8,
                         constraint=NULL) , stop=F)

allocs.G0.q1.5 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=0 , q=1.5, n=10000,minCV=8,
                         constraint=NULL) , stop=F)
allocs.G10.q1.5 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=10 , q=1.5, n=10000,minCV=8,
                         constraint=NULL) , stop=F)
allocs.G100.q1.5 <- all.allocations( par.vector= list( sigmasq.h=rep(1,length(CantP)) ,
                         rho.h=rep(0.025,length(CantP)) ,  Nh=CantP ,
                           minsamp=rep(1,length(CantP)), G=100 , q=1.5, n=10000,minCV=8,
                         constraint=NULL) , stop=F)

# Table 1: Efficiency for q=0, 0.5, 1, 1.5, 2 for G=0

table1 <- cbind( allocs.G0.q0$F.table , allocs.G0.q0.5$F.table ,allocs.G0.q1$F.table ,
                          allocs.G0.q1.5$F.table , allocs.G0.q2$F.table )[1:6,]
colnames(table1) <- paste("q=",c(0,0.5,1,1.5,2),sep="")
rownames(table1) <- c("Equal allocation" , "Proportional allocation" ,
                      "Optimal for composite" ,
                      "Optimal for composite with constraints",
                      "Optimal power allocation" , "Optimal power with constraints")
table1 <- round(table1, digits=3)
table1
table1.txt <- format(table1,digits=3,nsmall=3)
write.table(table1.txt,sep=" & " , eol="\\\\ \n" , quote=F , row.names=T,col.names=F,
            file="table1.txt")

# Table 2: Efficiency for q=0, 0.5, 1, 1.5, 2 for G=10

table2 <- cbind( allocs.G10.q0$F.table , allocs.G10.q0.5$F.table ,allocs.G10.q1$F.table ,
                          allocs.G10.q1.5$F.table , allocs.G10.q2$F.table )[1:6,]
colnames(table2) <- paste("q=",c(0,0.5,1,1.5,2),sep="")
rownames(table2) <- c("Equal allocation" , "Proportional allocation" ,
                      "Optimal for composite" ,
                      "Optimal for composite with constraints",
                      "Optimal power allocation" , "Optimal power with constraints")
table2 <- round(table2, digits=3)
table2.txt <- format(table2,digits=3,nsmall=3)
write.table(table2.txt,sep=" & " , eol="\\\\ \n" , quote=F , row.names=T,col.names=F,
            file="table2.txt")

# Table 3: Efficiency for q=0, 0.5, 1, 1.5, 2 for G=100
```

```
table3 <- cbind( allocs.G100.q0$F.table , allocs.G100.q0.5$F.table ,allocs.G100.q1$F.table ,
                          allocs.G100.q1.5$F.table , allocs.G100.q2$F.table )[1:6,]
colnames(table3) <- paste("q=",c(0,0.5,1,1.5,2),sep="")
rownames(table3) <- c("Equal allocation" , "Proportional allocation" ,
                      "Optimal for composite" ,
                      "Optimal for composite with constraints",
                      "Optimal power allocation" , "Optimal power with constraints")
table3 <- round(table3, digits=3)
table3.txt <- format(table3,digits=3,nsmall=3)
write.table(table3.txt,sep=" & " , eol="\\\\ \n" , quote=F , row.names=T,col.names=F,
            file="table3.txt")


# Table 4: numerical optima for power by q and G

table4 <- rbind( c( allocs.G0.q0$optimal.power , allocs.G10.q0$optimal.power ,
                             allocs.G100.q0$optimal.power ) ,
                 c( allocs.G0.q0.5$optimal.power , allocs.G10.q0.5$optimal.power ,
                     allocs.G100.q0.5$optimal.power ) ,
                 c( allocs.G0.q1$optimal.power , allocs.G10.q1$optimal.power ,
                     allocs.G100.q1$optimal.power ) ,
                 c( allocs.G0.q1.5$optimal.power , allocs.G10.q1.5$optimal.power ,
                     allocs.G100.q1.5$optimal.power ) ,
                 c( allocs.G0.q2$optimal.power , allocs.G10.q2$optimal.power ,
                     allocs.G100.q2$optimal.power ) )
colnames(table4) <- paste("G=",c(0,10,100),sep="")
rownames(table4) <- c("q=0","q=0.5","q=1","q=1.5","q=2")
table4 <- round(table4, digits=3)
table4
write.table(format(t(table4),digits=3,nsmall=3),sep=" & " , eol="\\\\ \n" , quote=F ,
                  row.names=T,col.names=F,file="table4.txt")

# Table 5: numerical optima for power by q and G with constraint

table5 <- rbind( c( allocs.G0.q0$optimal.power.constrained ,
                              allocs.G10.q0$optimal.power.constrained,
                              allocs.G100.q0$optimal.power.constrained ) ,
                 c( allocs.G0.q0.5$optimal.power.constrained ,
                     allocs.G10.q0.5$optimal.power.constrained ,
                     allocs.G100.q0.5$optimal.power.constrained ) ,
                 c( allocs.G0.q1$optimal.power.constrained ,
                     allocs.G10.q1$optimal.power.constrained ,
                     allocs.G100.q1$optimal.power.constrained ) ,
                 c( allocs.G0.q1.5$optimal.power.constrained ,
                     allocs.G10.q1.5$optimal.power.constrained ,
                     allocs.G100.q1.5$optimal.power.constrained ) ,
                 c( allocs.G0.q2$optimal.power.constrained ,
                     allocs.G10.q2$optimal.power.constrained ,
                     allocs.G100.q2$optimal.power.constrained ) )
colnames(table5) <- paste("G=",c(0,10,100),sep="")
rownames(table5) <- c("q=0","q=0.5","q=1","q=1.5","q=2")
table5 <- round(table5, digits=3)
table5
write.table(format(t(table5),digits=3,nsmall=3),sep=" & " , eol="\\\\ \n" , quote=F ,
                  row.names=T,col.names=F,file="table5.txt")


# Figure 1: Boxplots of Sample Size by q (0,1,2) and G (0,10,50)

figure1data <- data.frame( q=rep(c(0.5,1,1.5,2), each=26) ,
                           nh2=c( allocs.G0.q0.5$nh.table[,2] , allocs.G0.q1$nh.table[,2] ,
                                       allocs.G0.q1.5$nh.table[,2] , allocs.G0.q2$nh.table[,2] ) ,
                           nh3=c( allocs.G0.q0.5$nh.table[,3] , allocs.G0.q1$nh.table[,3] ,
                                       allocs.G0.q1.5$nh.table[,3] , allocs.G0.q2$nh.table[,3] ) ,
                           nh4=c( allocs.G0.q0.5$nh.table[,4] , allocs.G0.q1$nh.table[,4] ,
                                       allocs.G0.q1.5$nh.table[,4] , allocs.G0.q2$nh.table[,4] ) ,
                           nh5=c( allocs.G0.q0.5$nh.table[,5] , allocs.G0.q1$nh.table[,5] ,
                                       allocs.G0.q1.5$nh.table[,5] , allocs.G0.q2$nh.table[,5] ) ,
                           nh6=c( allocs.G0.q0.5$nh.table[,6] , allocs.G0.q1$nh.table[,6] ,
                                       allocs.G0.q1.5$nh.table[,6] , allocs.G0.q2$nh.table[,6] ) )
figure1data$CVpct2 <- sqrt( 0.025*(1-0.025) / (1+(figure1data$nh2-1)*0.025) ) * 100
figure1data$CVpct3 <- sqrt( 0.025*(1-0.025) / (1+(figure1data$nh3-1)*0.025) ) * 100
figure1data$CVpct4 <- sqrt( 0.025*(1-0.025) / (1+(figure1data$nh4-1)*0.025) ) * 100
figure1data$CVpct5 <- sqrt( 0.025*(1-0.025) / (1+(figure1data$nh5-1)*0.025) ) * 100
figure1data$CVpct6 <- sqrt( 0.025*(1-0.025) / (1+(figure1data$nh6-1)*0.025) ) * 100
postscript(file="figure1.eps",height=8,width=8,onefile=FALSE,horizontal=F,paper="special")
old.mar <- par()$mar
par(mfrow=c(2,2))
boxplot(CVpct3~q,data=figure1data,xlab="q",ylab="",ylim=c(0,16) ,
             main="Optimal for composite")
boxplot(CVpct4~q,data=figure1data,xlab="q",ylab="",ylim=c(0,16) ,
             main="Optimal for composite with constraints")
boxplot(CVpct5~q,data=figure1data,xlab="q",ylab="",ylim=c(0,16) ,
             main="Optimal power allocation")
boxplot(CVpct6~q,data=figure1data,xlab="q",ylab="",ylim=c(0,16) ,
             main="Optimal power allocation with constraints")
box("outer", lty="solid",col="black")
par(mar=old.mar)
dev.off()
```