International Journal of Computational and Theoretical Statistics
ISSN (2210-1519)
Int. J. Comp. Theo. Stat. 3, No. 2 (Nov-2016)

Smaller Balanced Sampling Plans Excluding Adjacent Units for One Dimensional Population

Rajeev Kumar[1], Rajender Parsad[1], and B. N. Mandal[1]

[1] *ICAR-Indian Agricultural Statistics Research Institute, Library Avenue, Pusa, New Delhi 110012, India*

Received January 30, 2016, Revised July 9, 2016, Accepted July 21, 2016, Published November 1, 2016

Abstract: Balanced sampling plans excluding adjacent units are used for selecting samples from naturally ordered populations where nearer units provide similar measurements. In this article, we present several new balanced sampling plans excluding adjacent units for one dimensional populations with circular and linear ordering of units. The plans are obtained through a new algorithm based on linear integer programming.

Keywords: Balanced sampling plans, Polygonal designs, Finite population sampling, Block designs, Algorithm, Integer programming

1. INTRODUCTION

In finite population sampling, we are concerned with the estimation of population parameters by selecting a sample of size n from a population containing N distinct identifiable units. Often the units in the population may be arranged in space or time and as a result, nearby population units may provide similar measurements and hence, it would be desirable to draw a sample which avoids the selection of adjacent units. Under one dimensional circular ordering of population units, [1] introduced a sampling plan under which the second order inclusion probability for any non-contiguous pair of units is constant and zero for any contiguous pair of units.

[2] extended these plans to balanced sampling plans excluding adjacent units under which the second order inclusion probability for any non-adjacent pair of units is constant and zero for any adjacent pair of units. Here, two units are adjacent whenever they are within a distance of α units from each other. Under one dimensional circular ordering, the distance between two units i and j is given by $\delta(i, j) = \text{Min}\{|i - j|, \nu - |i - j|\}$ and for linear ordering, the distance is $\delta(i, j) = \text{Max}\{i - j, j - i\}$, $i \neq j = 1, 2, ..., N$. We shall denote a balanced sampling plan excluding adjacent units for population size N, sample size n and distance α in general as BSA(N, n, α). A BSA(N, n, α) under circular and linear ordering of population units is denoted as cBSA(N, n, α) and lBSA(N, n, α), respectively.

There is a lot of interest in the existence and construction of BSA for given parameters N, n and α. Existence and construction of cBSA for $n = 2$ is completely solved by [1] for $\alpha = 1$ and by [2] for $\alpha > 1$. For $n = 3$, existence and construction results of cBSA are available in [3]; [4]; [5]; [6];[7];[8] and [9]. [10] and [11] provided solution of cBSA for $n = 4$ and $\alpha = 1$. [12]; [13] and [14] presented interesting existence results for cBSA for $n \geq 4$ and $\alpha \geq 1$. The following theorem gives an existence result.

Theorem 1. A necessary condition for existence of a cBSA(N, n, α) is $N \geq (2\alpha + 1)n$ for $n \geq 3$ and $\alpha \geq 1$ and $N \geq (2\alpha + 1)n + 1$ for the following combinations of (n, α) : $\{(n \geq 5, 1), (6 \leq n \leq 12, 2), (5 \leq n \leq 9, 3), (6 \leq n \leq 8, 4)$ and $(6 \leq n \leq 7, 5)\}$.

Proofs of the results in Theorem 1 can be found in [2], [12] and [13]. For certain parametric combinations, the necessary conditions are also sufficient.

Another result due to [2] is stated below.

E-mail address: rajeev20363@gmail.com, rajender.parsad@icar.gov.in (corresponding Author), mandal.stat@gmail.com

Theorem 2. The existence of a *c*BSA(N, n, α) implies the existence of a *c*BSA($N + 2\alpha + 1$, n, α).

[15]; [13]; [16] and [9] presented algorithms to construct *c*BSA(N, n, α) for small values of N. Theorem 2 then can be utilized to get bigger *c*BSA for bigger values of N. However, application of Theorem 2 on a *c*BSA(N, n, α) with support size b gives a *c*BSA($N + 2\alpha + 1$, n, α) with support size $(N + 2\alpha + 1)b$, see [2] for details.

But there may exist a *c*BSA($N + 2\alpha + 1$, n, α) with support size less than $(N + 2\alpha + 1)b$, which we shall see in Section 3.

[13] provided the following result to obtain *l*BSAs from *c*BSAs and vice versa.

Theorem 3. The existence of a *c*BSA(N, n, α) implies the existence of a *l*BSA($N - \alpha$, $n - 1$, α) and a *l*BSA($N - (\alpha + 1)$, $n - 1$, α). Further, the existence of a *l*BSA(N, n, α) implies the existence of a *c*BSA($N + \alpha$, n, α) and a *c*BSA($N + \alpha + 1$, n, α).

[13] and [16] presented linear programming approaches to obtain smaller *c*BSA which then can be utilized to obtain more *c*BSA and *l*BSAs. However, again application of Theorem 3 results in BSAs with bigger support sizes. Therefore, effort is required to identify BSAs with smaller support sizes. Moreover, most of the methods produce *c*BSAs which are cyclic in nature, i.e., the support of the plan can be obtained by cyclically developing initial generator samples modulo N. However, there may exist non-cyclic *c*BSAs with smaller support sizes for a given N, n and α and such BSAs need to be identified.

In this article, we present several new *c*BSA and *l*BSAs with smaller support sizes. We obtain these BSAs by developing an algorithm following [17] and [18]. An important feature of the proposed algorithm is that it can construct cyclic or non-cyclic *c*BSA and *l*BSAs. In Section 2, we present the algorithm in detail. Section 3 presents the *c*BSA and *l*BSAs obtained using the algorithm. We conclude the article in Section 4.

2. ALGORITHM FOR OBTAINING BSAs

The algorithm obtains the support of a BSA by constructing a polygonal design which has one to one correspondence with a BSA. A polygonal design is an arrangement of N symbols belonging to the set $\{1, 2, ..., N\}$ in b blocks such that each block has n distinct symbols, i^{th} ($i = 1, 2, ..., N$) symbol appears in r_i blocks and each pair of symbols which are at a distance more than α appear together in λ blocks and all other pairs do not appear together in any block. The entities N, b, r_i, n, λ and α are the parameters of the design. Under circular ordering of the symbols, $r_i = r\ \forall i$ and they satisfy the following necessary conditions:

$$Nr = bn$$

$$\lambda(N - 2\alpha - 1) = r(n - 1). \tag{1}$$

Under linear ordering of the symbols, [13] have shown that

$$r_i = \begin{cases} \dfrac{\lambda(N-\alpha-1)}{n-1} & \text{if} \quad 1 \le i \le \alpha \\ rN - i + 1 & \text{if} \quad N-\alpha+1 \le i \le N \\ \dfrac{\lambda(N-2\alpha-1)}{n-1} & \text{otherwise} \end{cases} \tag{2}$$

$$\lambda\,(N - \alpha)\,(N - \alpha - 1) = bn\,(n - 1).$$

If we consider the N symbols as sampling units, the blocks as samples, the symbols in a block as the units in the sample and then if every block of a polygonal design is given probability of selection as $1/b$, then the polygonal design is equivalent to a BSA(N, n, α). Thus, obtaining a polygonal design is equivalent to obtaining a BSA.

Since a polygonal design is an incomplete block design, it can be represented by a $N \times b$ incidence matrix $N = (n_{is})$ where n_{is} denotes number of times i^{th} ($i = 1, 2, ..., N$) symbol appears in s^{th} ($s = 1, 2, ..., b$) block. Clearly, $n_{is} \in \{0, 1\}\ \forall\ i$, s for a polygonal design. Given the parameters of a polygonal design, the proposed algorithm attempts to construct its incidence matrix by obtaining its rows one by one. In the first step, we obtain the first row by randomly filling r_1 positions with 1 and rest positions with 0. We obtain the i^{th} ($i = 2, 3, . . . , N$) row such that this row has 1 in r_i positions, the concurrence of this row with a row within a distance of α is zero and with a row with a distance of more than α is λ

and the s^{th} block size do not exceed n. We implement this i^{th} step through a linear integer programming (LIP) formulation. If we are able to get all the N rows, then our construction is complete. To make the exposition clearer, we detail the procedure of obtaining the i^{th} row.

Let the s^{th} (s = 1, 2, ..., b) block size after (i − 1) steps is k_s. Then we compute weights $w_s = \frac{1}{n_s}$ whenever $n_s > 0$, otherwise we set $w_s = 1$. Let the elements of the i^{th} row of the **N** matrix be $(x_1, x_2, \ldots, x_b)'$ and they are unknown. Then we solve the LIP problem with respect to binary decision variables $x_1, x_2, \ldots, x_b$:

Maximize $\phi = \sum_{s=1}^{b} w_s r_s$

Subject to constraints

$$\sum_{s=1}^{b} x_s = r_i \tag{3}$$

$$\sum_{s=1}^{b} n_{js} x_s = \begin{cases} \lambda, & \text{if } \delta(i,j) > 0 \\ & \text{and } j = 1,2,\ldots,i-1 \\ 0 & \text{if } \delta(i,j) \le \alpha \end{cases}$$

$$x_s \le n - n_s \forall\, s = 1,2,\ldots,b.$$

If an optimal solution to the LIP problem (3) exists, then the solution gives the i^{th} row of the incidence matrix. Then we proceed to obtain the next row of the incidence matrix. However, sometimes the LIP problem (3) may not have a feasible solution and in that scenario, we delete one row at random from the rows 1 to (i − 1) and store the deleted row in a matrix **T** and update n_s. Let the m^{th} row be deleted where $1 \le m \le (i-1)$. We then try to obtain an alternative solution for the m^{th} row of the incidence matrix. For this, we solve the following LIP formulation:

Maximize $\phi = \sum_{s=1}^{b} w_s r_s$

Subject to constraints

$$\sum_{s=1}^{b} x_s = r_m \tag{4}$$

$$\sum_{s=1}^{b} n_{js} x_s = \begin{cases} \lambda, & \text{if } \delta(m,j) > 0 \\ & \text{and } j = 1,2,\ldots,m-1,m+1,\ldots,i-1 \\ 0 & \text{if } \delta(m,j) \le \alpha \end{cases}$$

$$x_s \le n - n_s \forall\, s = 1,2,\ldots,b.$$

$$\sum_{s=1}^{b} t_{qs} r_s < r_{m,q} = 1,2,\ldots,p.$$

In the formulation (4), t_{qs} denote the element in the q^{th} row and s^{th} column of **T** matrix. The last constraint in the formulation (4) ensures that the deleted rows stored in **T** do not appear again as solution. An optimal solution to the formulation (4) gives an alternate solution for the m^{th} row of the incidence matrix. If the formulation (4) does not have a feasible solution, we try deleting another row. Once a solution for the m^{th} row is obtained, we proceed to obtain the i^{th} row as before using formulation (3). We stop when all the N rows of the incidence matrix are obtained.

Remark 1. When we do not get a feasible solution for (3), there might be two reasons for this: (i) some wrong candidate rows might have been selected in the previous steps, however, it is not known which row is a wrong candidate row or (ii) no designs exists for the parametric setting. In this algorithm, we are not able to conform situation of type (ii). Hence, we can try to eliminate one of the bad rows which are selected up to (i-1) rows so that we can get a feasible solution for the i^{th} row. So we suggest deleting one row at random from the 1 to (i-1) rows and try to get an alternative row for the deleted row by formulation (4).

Remark 2. Sometimes we are not able to get solution beyond a particular row even after solving formulation (4). In that case, we restart the algorithm and call it another 'trial'.

We give a brief sketch of the algorithm below.

The algorithm for obtaining BSA

Input: Input parameters.
Step 1 Obtain first row of **N** at random with r_1 elements as 1 and rest as 0. Calculate n_s and w_s, $s = 1, 2, ..., b$.

Step *i* For obtaining i^{th} row ($i = 2, 3, ..., N$), solve the formulation (3).

- If solution of (3) exists, compute n_s and w_s, $s = 1, 2, ..., b$ and go to next i.
- If solution of (3) does not exist, repeat the following step

  -- delete a random m^{th} row where $1 \leq m \leq i - 1$, compute n_s and w_s, $s = 1, 2, ..., b$ and solve the formulation (4) until a solution for m^{th} row is found.

Output: Output **N**.

Remark 3. Note that the algorithm does not guarantee that a BSA will be obtained even if a BSA exists for a given parametric combination. However, from our experience, we noted that the algorithm is able to obtain most of the existent BSAs with smaller population size. The performance of the algorithm goes down for BSAs with population size greater than 40. This is due to the fact that the chances of entering wrong candidate rows in the incidence matrix go up with the increase in the number of rows of the incidence matrix. With respect to computation time, the algorithm takes very less time to generate a BSA, for example, to construct a *c*BSA(20,5,1), it took 2.5 elapsed seconds on a Intel Core i5 3.20 GHz CPU with 8GB RAM and 64 bit machine with Windows 7 operating system.

3. CONSTRUCTION OF BSAs USING THE PROPOSED ALGORITHM

We have used the algorithm 2 to construct *c*BSA and *l*BSA in the parametric range $N \leq 30$, $n \leq 5$, $\lambda \leq 5$, $\alpha \leq 5$. In the above parametric range, a total of 1037 parametric combinations satisfy the necessary conditions (1). Distribution of these 1037 parametric combinations for α = 1, 2, 3, 4 and 5 along with the number of designs obtained through the algorithm, the number of non-existent designs, the number of designs for which either the solution is unknown or non-existence is not proved and the number of new designs obtained is given in Table 1.

Table 1: Distribution of parametric combinations in the range $N \leq 30$, $n \leq 5$, $\lambda \leq 5$, $\alpha \leq 5$ for circular ordering of units

	α = 1	α = 2	α = 3	α = 4	α = 5	Total
Number of parametric combinations	233	231	214	182	177	1037
Number of designs obtained	216	191	143	112	94	756
Number of non-existing designs	17	40	70	70	83	280
Number of designs for which solution is unknown	0	0	1	0	0	1
Number of new designs	1	0	4	0	0	5

Table 2: New *c*BSAs in the range $N \leq 30$, $n \leq 5$, $\alpha \leq 5$

Sl. No.	N	b	r	n	λ	α	Remarks
1	20	68	17	5	4	1	
2	21	49	7	3	1	3	
3	21	98	14	3	2	3	2 copies of design at Sr. No. 2
4	21	196	28	3	4	3	4 copies of design at Sr. No. 2
5	21	245	35	3	5	3	5 copies of design at Sr. No. 2

From Table 1, we see that out of 1037 parametric combinations, designs are non-existent for 280 combinations due to Theorems of [13] and Result 2.1 of [19]. In the parametric range considered, we could not obtain a solution for *c*BSA(28,4,3) with a support size 49. Out of 756 designs obtained, 5 designs are new in the sense that their solution was not available in the literature earlier. The parameters of 5 new designs are given in Table 2.

As mentioned earlier, the proposed algorithm can generate BSAs which may be cyclic or non-cyclic. Most of the *c*BSAs in the literature are cyclic in nature and the support sizes of the existent cyclic *c*BSAs are bigger. In Table 2, all the new *c*BSAs are non-cyclic in nature, and the support sizes of these plans are smaller than corresponding existing cyclic BSAs in literature. For example, for $N = 20$, $n = 5$, $\alpha = 1$, a cyclic *c*BSA of [16] has a support size of 280, whereas here we have a support size of 68 in Sl. No. 1 of Table 2 and for $N = 21$, $n = 3$, $\alpha = 3$, *c*BSA of [2] has a support size of 147 whereas the plan in Sl. No. 2 of Table 2 has a support size of only 49.

To obtain *l*BSAs, we found that a total of 817 parametric combinations satisfy necessary conditions (2) in the parametric range $N \leq 30$, $n \leq 5$, $\lambda \leq 5$, $\alpha \leq 5$. Distribution of these 817 parametric combinations for α = 1, 2, 3, 4 and 5 along with the number of designs obtained through the algorithm, the number of non-existent designs, the number of designs for which either the solution is unknown or non-existence is not proved and the number of new designs obtained is given in Table 3.

Table 3: Distribution of parametric combinations in the range N ≤ 30, n ≤ 5, λ ≤ 5, α ≤ 5 for linear ordering of units

	α = 1	α = 2	α = 3	α = 4	α = 5	Total
Number of parametric combinations	188	177	163	151	138	817
Number of designs obtained	178	152	125	107	95	657
Number of non-existing designs	10	25	38	44	43	160
Number of new designs	7	7	0	2	0	16

From Table 3, we see that out of 817 designs, 160 designs are non-existent due to Theorems 6.1 and 6.2 and Table 7 of [13]. Out of 657 designs obtained, 16 designs are new in the sense that their solution was not available in the literature earlier.
The parameters of 16 new designs are given in Table 4.

Table 4: New *l*BSAs in the range $N \leq 30$, $n \leq 5$, $\lambda \leq 5$, $\alpha \leq 5$

Sl. No.	N	b	r_1	r_2	n	λ	α
1	21	76	19	18	5	4	1
2	22	84	20	19	5	4	1
3	25	138	23	22	4	3	1
4	26	120	24	23	5	4	1
5	27	130	25	24	5	4	1
6	29	189	27	26	4	3	1
7	30	203	28	27	4	3	1

Sl. No.	N	b	r_1	r_2	r_3	n	λ	α
8	22	95	19	18	17	4	3	2
9	23	105	20	19	18	4	3	2
10	26	138	23	22	21	4	3	2
11	27	120	24	23	22	5	4	2
12	27	150	24	23	22	4	3	2
13	28	130	25	24	23	5	4	2
14	30	189	27	26	25	4	3	2

Sl. No.	N	b	r_1	r_2	r_3	r_4	r_5	n	λ	α	Remark
15	29	200	24	23	22	21	20	3	2	4	
16	29	400	48	46	44	42	40	3	4	4	2 copies of design at Sr. No. 15

Though we have used the proposed algorithm to obtain BSAs in the restricted parametric range $N \leq 30$, $n \leq 5$, $\lambda \leq 5$, $\alpha \leq 5$, the algorithm is general in nature and can be used for obtaining BSAs outside this parametric range. However, we have seen that bigger value of N and n increases the chance of getting inappropriate candidate rows for the incidence matrix and hence, lower the performance of the algorithm to get the desired designs.

4. CONCLUDING REMARKS

In this article, we have proposed an algorithm to obtain balanced sampling plans excluding adjacent units for circular and linear ordering of population units. We have obtained a number of circular and linear BSAs using the proposed algorithm in the parametric range $N \leq 30$, $n \leq 5$, $\lambda \leq 5$, $\alpha \leq 5$. We found several new circular and linear BSAs in the above parametric range. The number of plans for which solution is unknown is 1 in case of circular ordering of population units. Further research efforts are required to either obtain the plan or to prove its non-existence. For large population and/or sample sizes, the algorithm may not be very effective. Therefore, developing methodology to obtain BSAs for large population and sample sizes would be an interesting area of research.

ACKNOWLEDGEMENTS

This article is a part of M.Sc. thesis work of Rajeev Kumar (2015) carried out in ICAR-Indian Agricultural Research Institute, New Delhi, India. The authors are thankful to the anonymous reviewers and Editor for their useful suggestions that have resulted in better presentation of the results

REFERENCES

[1] A.S. Hedayat, C.R. Rao, and J. Stufken, “Sampling plans excluding contiguous units,” Journal of Statistical Planning and Inference, vol. 19(2), pp. 159–170, 1988.

[2] J. Stufken, “Combinatorial and statistical aspects of sampling plans to avoid the selection of adjacent units,” Journal of Combinatorics Information and System Science, vol. 18, pp. 81–92, 1993.

[3] C.J. Colbourn and A.C. Ling, “A class of partial triple systems with applications in survey sampling,” Communications in Statistics-Theory and Methods , vol. 27(4), pp. 1009–1018, 1998.

[4] R. Wei, “Cyclic BSEC of block size 3,” Discrete Mathematics , vol. 250(1), pp. 291–298, 2002.

[5] J. Zhang, “Bicyclic BSECs of block size 3 and any λ,” Discrete Mathematics , vol. 313(15), pp. 1586–1588, 2013.

[6] J. Zhang and Y. Chang, “The spectrum of cyclic BSA (ν, 3, λ; α) with α= 2, 3,” Journal of Combinatorial Designs, vol. 13(5), pp. 313–335, 2005a.

[7] J. Zhang and Y. Chang, “The spectrum of cyclic BSEC with block size three,” Discrete Mathematics, vol. 305(1), pp. 312–322, 2005b.

[8] J. Zhang, and Y. Chang, “Existence of BSAs and cyclic BSAs of block size three,” Journal of Statistical Planning and Inference, vol. 137(4), pp. 1496–1511, 2007.

[9] B.N. Mandal, V.K. Gupta, and R. Parsad, “Construction of efficient mixed-level k-circulant supersaturated designs,” Journal of Statistical Theory and Practice, vol. 5(4), pp. 627– 648, 2011.

[10] C.J. Colbourn and A.C. Ling, “Balanced sampling plans with block size four excluding contiguous units,” Australasian Journal of Combinatorics , vol. 20, pp. 37–46, 1999.

[11] J. Zhang, and Y. Chang, “Partitionable sets and cyclic BSECs with block size four,” Journal of Statistical Planning and Inference, vol. 139(6), pp. 1974–1979, 2009.

[12] J. Stufken, S. Y. Song, K. See, and K. R. Driessel, “Polygonal designs: Some existence and non-existence results,” Journal of Statistical Planning and Inference, vol. 77(1), pp. 155– 166, 1999.

[13] J.H. Wright and J. Stufken, “New balanced sampling plans excluding adjacent units,” Journal of Statistical Planning and Inference, vol. 138(11), pp. 3326–3335, 2008.

[14] P.J. Dukes and A.C.H. Ling, “Existence of balanced sampling plans avoiding cyclic distances,” Metrika, vol. 70(2), pp.131–140, 2009.

[15] J. Stufken and J.H. Wright, “Polygonal designs with blocks of size k ≤ 10,” Metrika, vol. 54(2), pp.179–184, 2001.

[16] B.N. Mandal, R. Parsad, and V.K. Gupta, “Computer-aided construction of balanced sampling plans excluding contiguous units,” Journal of Statistics and Applications, vol. 3, pp. 1-2, 2008.

[17] B.N. Mandal, "Linear integer programming approach to construction of balanced incomplete block designs," Communications in Statistics-Simulation and Computation, vol. 44(6), pp. 1405–1411, 2015.

[18] B.N. Mandal, V.K. Gupta, and R. Parsad, "Efficient incomplete block designs through linear integer programming," American Journal of Mathematical and Management Sciences , vol. 33(2), pp. 110–124, 2014.

[19] R. Parsad, S. Kageyama, and V.K. Gupta, "Use of complementary property of block designs in PBIB designs," Ars Combinatoria, vol. 85, pp. 173–182, 2007.