Available online at www.isas.org.in/jisas
JOURNAL OF THE INDIAN SOCIETY OF AGRICULTURAL STATISTICS 64(2) 2010 303-312

Linear Integer Programming Approach to Construct Distance Balanced Sampling Plans

B.N. Mandal[1], Rajender Parsad[2] and V.K. Gupta[2*]
[1]*Indian Institute of Soil Science, Bhopal*
[2]*Indian Agricultural Statistics Research Institute, New Delhi*

Received 27 April 2010; Revised 30 June 2010; Accepted 18 July 2010

SUMMARY

Distance balanced sampling plans (DBSP) are a class of sampling plans in which second order inclusion probabilities are non-decreasing function of the distance between the population units. DBSP were introduced by Mandal *et al.* (2009) as a generalization of balanced sampling plans excluding adjacent units. In this article, a general *w*-point DBSP (w = 1, 2, …, $\left[\frac{N}{2}\right]$, where N is the population size and $[x]$ denotes largest integer contained in x) is introduced and a method of construction of *w*-point DBSP using linear integer programming is proposed. The method is general in nature and two-point, three-point, $\left[\frac{N}{2}\right]$-point, many other DBSPs, simple random sampling without replacement, balanced sampling plans excluding contiguous units (Hedayat *et al*., 1988) and balanced sampling plans excluding adjacent units (Stufken, 1993) fall out as a particular case. A list of $\left[\frac{N}{2}\right]$-point DBSP for sample size three is obtained for population size $N \le 100$, where N is odd.

Keywords: Balanced sampling plans, Distance balanced sampling plans, Distance balanced incomplete block designs, Linear integer programming.

1. INTRODUCTION

Consider the problem of obtaining a fixed sample size n from a finite population U of N distinct and identifiable units, labeled as 1, 2, ..., N. For mathematical convenience and expositive simplicity, we assume circular ordering of the units. Under circular ordering of units, the distance between a pair of units i and j is given by $\delta(i,j) = \min\{|i-j|, N-|i-j|\}$, $\forall\ i \neq j \in U$. Further, $\max_{i \neq j \in U} \delta(i,j) = d = [N/2]$. We shall let m denote the distances $\delta(i,j)$. Then m can take values 1, 2, ..., d. In survey sampling, there arise situations when the population units are ordered in time or space. This natural ordering of the units induces some positive correlation between the nearer units and hence contiguous units in the population provide similar observations. Hedayat *et al.* (1988) introduced balanced sampling plans excluding contiguous units (BSEC) for such situations. BSECs are those sampling plans in which the pairs of contiguous units do not appear together in a sample whereas all other pairs appear equally often in the samples. Stufken (1993) generalized the concept of BSEC to Balanced Sampling Plans Excluding Adjacent units [BSA(m) plan] by excluding all those pairs of units from the sample whose distance is less than or equal to m. Two units are called adjacent when their distance is less than a specified number m.

Although BSA(m) plans are appealing for situations when the population units are distributed over

* *Corresponding author* : V.K. Gupta
E-mail address : vkgupta@iasri.res.in; vkgupta_1751@yahoo.co.in

time or space, it is not possible to obtain an unbiased estimator of variance of the Horvitz-Thompson (1952) estimator of population mean under these plans because all possible pairs of population units do not have a non-zero second order inclusion probability. To overcome this problem, Mandal *et al.* (2009) introduced a family of sampling plans called distance balanced sampling plans (DBSP) which is defined in the sequel.

Definition 1.1. DBSPs are sampling plans in which the first order inclusion probabilities are constant for all the population units and second order inclusion probabilities are non-decreasing function of the distance between the population units.

More explicitly, for given N and n, the first and second order inclusion probabilities are given by

$$\pi_i = \frac{n}{N}, \; i = 1, 2, ..., N$$

and $$\pi_{ij} = \frac{n(n-1)}{N} d_{ij}, \; i \neq j = 1, 2, ..., N \qquad (1.1)$$

where $d_{ij} = \dfrac{f_{ij}}{\sum_{j(\neq i)=1}^{N} f_{ij}}$ and f_{ij} is a non-decreasing function of distance between units i and j

A DBSP for population size N, sample size n with distance function f_{ij} may be denoted as DBSP (N, n) with specification of f_{ij}.

Mandal *et al*. (2009) investigated three particular cases wherein f_{ij} takes two distinct values, three distinct values and d distinct values, respectively and called these as two-point, three-point and d-point DBSP.

Mandal *et al.* (2009) introduced distance balanced incomplete block designs (DSBIB) designs for obtaining DBSPs.

Definition 1.2. A DSBIB design is an arrangement of v treatments in b blocks of size k ($< v$) with r replications in such a way that

(i) each treatment $i \in \{1, 2, ..., v\}$ occurs at most once in a block

(ii) each block has k treatments

(iii) any two treatments (i, j) occur together in λ_{ij} blocks

(iv) λ_{ij}'s are non-decreasing function of $\delta(i, j)$, the distance between treatments i and j

(v) any two treatments (i, j) with same distance $\delta(i, j) = m$ have same concurrences $\lambda_{ij} = \lambda_m$, say.

The condition (v) signifies the name *distance balanced*. The parameters of DSBIB design satisfy the following relations:

(i) $vr = bk$

(ii) $$r(k-1) = \sum_{j(\neq i)=1}^{N} \lambda_{ij} \qquad (1.2)$$

These conditions are necessary but not sufficient.

Under circular population structure the maximum distance between two units is d, and, therefore, DSBIB design can have at most d distinct values of λ_{ij}'s namely $\lambda_1, \lambda_2, ..., \lambda_d$, satisfying $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_d$. A DSBIB design may be represented as ($v, b, r, k, \lambda_1, \lambda_2, ..., \lambda_d$). Considering $v = N$ and $k = n$, if the blocks of a DSBIB design are selected with probability $\frac{1}{b}$, then the design reduces to a DBSP. Henceforth, a DSBIB design with v treatments and block size k is synonymous with a DBSP with N units and sample size n.

Mandal *et al.* (2010) have utilized linear integer programming approach to construct BSA(m) plans. The purpose of this article is to introduce a general w-point DBSP ($w = 1, 2, \ldots, d$) and to propose a linear integer programming approach to obtain such w-point DBSPs.

The article is organized as follows. In Section 2, we introduce w-point DBSP ($w = 1, 2, \ldots, d$) and study its properties. In Section 3, we propose the linear integer programming approach for constructing w-point DBSPs and provide some examples. Existence conditions of w-point DBSPs and construction of d-point DBSPs for sample size three with $N \leq 100$, N odd, using the proposed approach are discussed in Section 4. In Section 5, the article is concluded with a discussion on future scope of research in this direction.

2. GENERAL *w*-POINT DISTANCE BALANCED SAMPLING PLANS

In this section, theory of a w-point DBSP is developed for a general choice of distance function between the units. Here, $w = 1, 2, \ldots, d$. Let the distance function be given by

$f_{ij} = f_{t+1}$ if $m_t < \delta(i, j) \le m_{t+1}$, $\forall i \neq j = 1, 2, ..., N$ and $t = 0, 1, 2, ..., w - 1$. Here $m_0 = 0$ and $m_w = d$ and f_z ($z = 1, 2, ..., w$) are non-negative integers such that $f_{t+1} \ge f_t$ for $t = 0, 1, 2, ..., w - 1$. (2.1)

With proper choice of w, the above defined distance function can produce SRSWOR plan, two-point, three-point, four-point, and in an extreme case, d-point DBSP. In other words, for $w = 1$, the plan reduces to SRSWOR; for $w = 2$, the plan reduces to two-point DBSP; for $w = 3$, the plan becomes three-point DBSP; and so on. For $w = d$, the plan reduces to d-point DBSP. Also, if $w = 2, f_1 = 0, f_2 = f$ and $m_1 = m$ then the plan reduces to BSA(m) plan.

Obviously, the DBSP described here produces a family of plans depending upon the choice of w, m_t and f_{ij}. The survey statistician based upon the knowledge of the population to be surveyed and practical convenience would decide about the member of the family of DBSPs to be used for given values of w, m_t and f_{ij}. Some previous knowledge about the population and the nature of the study variable would be a guiding force for the choice of a sampling plan. Different sampling plans that can be generated, are not the competitors because one has to judiciously decide the sampling plan depending upon the values of w and m and then the f_{ij}'s. Once the choice of a sampling plan is made, then its optimality property could be studied in terms of the variance of the Horvitz-Thompson estimator of the population total. The choice of the f_{ij}'s then assumes importance. Some guidelines about this choice are provided in the sequel wherein we study the properties of the w-point DBSPs.

For a given unit i, $i = 1, 2, ..., N$

$$\sum_{j(\neq i)=1}^{N} f_{ij} = 2\sum_{t=0}^{w-2}(m_{t+1} - m_t) f_{t+1} + (N - 2m_{w-1} - 1) f_w$$
$$= T \quad \text{(say)} \tag{2.2}$$

Therefore, the first and second order inclusion probabilities are given by

$$\pi_i = \frac{n}{N}, \; i = 1, 2, ..., N$$

and $$\pi_{ij} = \frac{n(n-1) f_{ij}}{NT}, \; i \neq j = 1, 2, ..., N$$

Hence,

$$\pi_{ij} = \frac{n(n-1) f_{t+1}}{NT} \text{ if } m_t < \delta(i, j) \le m_{t+1},$$
$$i \neq j = 1, 2, ..., N;\; t = 0, 1, 2, ..., w - 1. \tag{2.3}$$

The variance of Horvitz-Thompson estimator of population mean is given by

$$V(\hat{\bar{Y}}_{HT}) = \sigma^2[1 - \frac{(n-1)}{nT}\{2(f_1 - f_w)(m_1 - \sum_{j=1}^{m_1}\rho_j)$$
$$+2(f_2 - f_w)(m_2 - m_1 - \sum_{j=m_1+1}^{m_2}\rho_j) + \ldots$$
$$+2(f_{w-1} - f_w)(m_{w-1} - m_{w-2}$$
$$- \sum_{j=m_{w-2}+1}^{m_{w-1}}\rho_j) + f_w N\}] \tag{2.4}$$

It may be seen from (2.4) that the variance of Horvitz-Thompson estimator of population mean for a w-point DBSP depends on choices of $f_1, f_2, ..., f_w$. We illustrate for a particular case of $w = 2$ how the choice of f_{ij}'s is important for evaluating alternative plans. Consider two DBSPs as (i) DBSP (N, n) with $f_{ij} = f_1$, if $0 < \delta(i, j) \le m$ and $f_{ij} = f_2$, if $\delta(i, j) > m$ and (ii) DBSP (N, n) with $f_{ij} = f_1^*$, $0 < \delta(i, j) \le m$ and $f_{ij} = f_2^*$, if $\delta(i, j) > m$. Then the variance of Horvitz-Thompson estimator in case of DBSP (N, n, m, f_1, f_2) will be smaller than that for DBSP (N, n, m, f_1^*, f_2^*) if $\frac{f_1}{f_2} < \frac{f_1^*}{f_2^*}$.

Similar kinds of conditions may be derived for three or higher point DBSPs. Therefore, in a given situation it would be better to select a DBSP with smaller ratios of $f_1, f_2, ..., f_w$.

To obtain w-point DBSP using DSBIB designs, construct a DSBIB design with parameters v, b, k, r, $\lambda_1\mathbf{1}'_{a_1}, \lambda_2\mathbf{1}'_{a_2}, ..., \lambda_w\mathbf{1}'_{a_w}$, where $a_1 + a_2 + ... + a_w = \left[\frac{v}{2}\right] = d$ and $\mathbf{1}'_{a_i}$ denotes a_i-component row vector of ones, $i = 1, 2, \ldots, w$. It may be seen that for a w-point DSBIB design, $a_{t+1} = m_{t+1} - m_t$, $t = 0, 1, ..., w-1$. For example, for a two-point (*i.e.* $w = 2$) DSBIB design, $a_1 = m_1 = m$, $a_2 = d - m$; for a three-point (*i.e.* $w = 3$) DSBIB design $a_1 = m_1$, $a_2 = m_2 - m_1$, $a_3 = d - m_2$ and for a d-point DSBIB design, $a_i = 1\ \forall i$. For BIB (*i.e.* $w = 1$) design, $a_1 = d$.

3. LINEAR INTEGER PROGRAMMING APPROACH FOR CONSTRUCTION OF *w*-POINT CYCLIC DBSP PLANS

In this Section we describe a linear integer programming approach to generate w-point ($w = 1, 2, \ldots, d$) cyclic DBSPs or DSBIBs for given ($v, b, r, k, \lambda_1\mathbf{1}'_{a_1}, \lambda_2\mathbf{1}'_{a_2}, ..., \lambda_w\mathbf{1}'_{a_w}$) .

Step 1. For given v and k, generate all the possible blocks of size k ($<v$) of distinct treatments out of v treatments such that the first treatment in each block is 1. The total number of such blocks possible is $\omega = \binom{v-1}{k-1}$. From the k treatments in a block, obtain the $\frac{k(k-1)}{2}$ possible pair wise distances $\delta(i, j)$ between pair of treatments i and j, $i \neq j = 1, 2, ..., k$. The computed distances for a given block would belong to the set $\{1, 2, ..., d\}$. This would hold for all the ω blocks.

Step 2. Store the frequency of occurrences of distances in the ω blocks as ω incidence vectors. An incidence vector is a $d \times 1$ vector with elements as the frequencies of occurrences of the distances $1, 2, \ldots, d$, respectively, obtained in Step 1. The sum of all the elements of any incidence vector would be $\frac{k(k-1)}{2}$.

For clear exposition of the concept of incidence vectors, consider an example with $v = 7$, $k = 4$. Obviously $\omega = 20$. Consider one block out of 20 with block contents as (1, 2, 4, 7). The $\frac{k(k-1)}{2} = 6$ distances generated are $\delta(1, 2) = 1$, $\delta(1, 4) = 3$, $\delta(1, 7) = 1$, $\delta(2, 4) = 2$, $\delta(2, 7) = 2$ and $\delta(4, 7) = 3$ and the corresponding incidence vector is $(2, 2, 2)'$ with row sum as 6.

Step 3. From the ω incidence vectors generated at Step 2, retain only distinct incidence vectors and corresponding blocks that generate these incidence vectors. Here, two vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be distinct if $\mathbf{x} - \mathbf{y} \neq \mathbf{0}$. Let the number of such incidence vectors obtained, along with their corresponding generating blocks, be c ($\leq \omega$). Denote by $\mathbf{A}$ the matrix of order $d \times c$, with columns as the c distinct incidence vectors.

Obviously, $\mathbf{1}'\mathbf{A} = \frac{k(k-1)}{2}\mathbf{1}'$, where $\mathbf{1}$ is a vector of ones. It may not be possible to give an expression for obtaining the number c for given v and k.

Step 4. Solve the following linear integer programming problem for obtaining a cyclic DSBIB design ($v, b, r, k, \lambda_1\mathbf{1}'_{a_1}, \lambda_2\mathbf{1}'_{a_2}, ..., \lambda_w\mathbf{1}'_{a_w}$):

Let $\mathbf{x} = (x_1, x_2, ..., x_c)'$, $x_1, x_2, ..., x_c$ being unknown non-negative integers.

Minimize $\phi = \mathbf{1}'\mathbf{x}$ with respect to variables $x_1, x_2, ..., x_c$ subject to constraints

$$\text{(i)}\quad \mathbf{A}\mathbf{x} = \Lambda$$

$$\text{(ii)}\quad \mathbf{1}'\mathbf{x} = \frac{b}{v}. \tag{3.1}$$

where Λ is a d-component vector, with $\Lambda = (\lambda_1\mathbf{1}'_{a_1}, \lambda_2\mathbf{1}'_{a_2}, ..., \lambda_w\mathbf{1}'_{a_w})'$, if v is odd and $\Lambda = (\lambda_1\mathbf{1}'_{a_1}, \lambda_2\mathbf{1}'_{a_2}, ..., (\lambda_w - 1)\mathbf{1}'_{a_w}, \frac{\lambda_w}{2})'$, if v is even.

It may be seen that number of constraints is $d + 1$. Also note that only integral values of $\frac{b}{v}$ are accepted, which is evident from constraint (ii).

If there exists a cyclic DSBIB design ($v, b, r, k, \lambda_1\mathbf{1}'_{a_1}, \lambda_2\mathbf{1}'_{a_2}, ..., \lambda_w\mathbf{1}'_{a_w}$), then the optimal solution to (3.1) results in a vector of non-negative integer values, say, $\mathbf{x}_0 = (t_1, t_2, ..., t_c)'$. Since c is generally far greater than $\frac{b}{v}$, there will be at most $\frac{b}{v}$ non-zero t_i's and thus only at most $\frac{b}{v}$ columns of the matrix **A** are identified as required incidence vectors with number of repeats of incidence vectors being given by corresponding t_i's. Hence, the optimal solution of linear integer programming formulation (3.1) directly identifies the required incidence vectors (and corresponding combinations as generator blocks).

Remark 3.1. It may be noted that generated DSBIBs are cyclic DSBIBs. An incomplete block design with v treatments is said to be cyclic if the design is obtained by cyclical development of some initial generator blocks modulo v.

We illustrate the construction of a DSBIB design using linear integer programming approach in the sequel.

Example 3.1. Consider construction of a d-point DSBIB with parameters $v = 11$, $k = 3$, $\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = 3$, $\lambda_4 = 4$, $\lambda_5 = 5$. For this case $a_i = 1 \,\forall i$. From parametric conditions given in Section 4, we see that a cyclic DSBIB design may exist with above parameters for $b = 55$ and $r = 15$. In step 1 the algorithm generates 45 lexicographic combinations of 3 units out of 11 units. The combinations generated are (1, 2, 3), (1, 2, 4), ..., (1, 10, 11). In step 2, the algorithm generates the corresponding incidence vectors for the combinations and these are (2, 1, 0, 0, 0)′, (1, 1, 1, 0, 0)′, ..., (2, 1, 0, 0, 0)′. After removing the duplicate incidence vectors, 10 distinct incidence vectors are retained in step 3. Here $\omega = 45$, but $c = 10$. The distinct incidence vectors are stored in matrix **A** of order 5 × 10 and is given by

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 2 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 2 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

We need to identify a maximum of $\frac{b}{v} = \frac{r}{k} = \frac{2.\sum_{i=1}^{5}\lambda_i}{k(k-1)}$ = 5 distinct generator vectors out of 10 columns in **A**. The number of distinct generator vectors may be less than b/v, but in this case some generating vectors would have multiplicities. One way is to search $\binom{c}{b/v} = \binom{10}{5}$ combinations and find out the five required vectors, which is time consuming for larger c and $\frac{b}{v}$.

Use of linear integer programming approach in (3.1), as described in step 4, gives the solution: $\mathbf{x}_0 =$ (0 0 0 0 1 0 0 2 1 1)′ which means that 5th, 8th, 9th and 10th column of matrix **A** are the required incidence vectors and corresponding generator blocks are (1, 2, 7), (1, 3, 7), (1, 4, 7) and (1, 4, 8) with number of repetitions of each generator block as 1, 2, 1, 1 respectively. Hence the solution to DSBIB (11, 55, 15, 3, 1, 2, 3, 4, 5) is obtained by developing the generator blocks stated above modulo (11).

Example 3.2. Let us consider constructing a two-point DSBIB design for $v = 13$, $k = 4$, $\lambda_1 = 3$, $\lambda_2 = 6$, and $m = 2$. Here, $a_1 = m = 2$, $a_2 = d - m = 4$. From parametric conditions stated in Section 4, we get $b = 65$, $r = 20$. Here, $\omega = 220$, but $c = 34$. Also, Λ is given by $\Lambda = (3, 3, 6, 6, 6, 6)'$. Linear integer programming formulation (3.1) gives a solution $\mathbf{x}_0 =$ (0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 1)′ which identifies 8th, 21st, 22nd, 32nd and 34th column of matrix **A** as the required incidence vectors and the corresponding generator blocks are (1, 2, 4, 7), (1, 2, 6, 9), (1, 2, 6, 10), (1, 3, 7, 9) and (1, 4, 7, 10) with number of repetitions of each generator block as 1.

Example 3.3. Now, we will construct a three-point DSBIB design for $v = 16$, $k = 4$, $\lambda_1 = 4$, $\lambda_2 = 8$, $\lambda_3 = 12$, $m_1 = 2$ and $m_2 = 4$. Here, $a_1 = m_1 = 2$, $a_2 = m_2 - m_1 = 2$, $a_3 = d - m_2 = 4$. For this case, $b = 176$, $r = 44$. We found, $\omega = 455$ and $c = 70$. Here, $\Lambda = (4, 4, 8, 8, 12, 12, 12, 6)'$. Linear integer programming formulation provides us the generator blocks (1, 2, 8, 11), (1, 3, 7, 12), (1, 4, 9, 12) and (1, 5, 9, 13) with number of repetitions as 4, 4, 2 and 1 respectively.

Example 3.4. Consider constructing a BIB design with $v = 21$, $k = 5$, $\lambda = 2$. We get $b = 42$ and $r = 10$. Here, $u = 1$ and $a_1 = \left\lfloor \frac{v}{2} \right\rfloor = 10$ and $\Lambda = (2,2,2,2,2,2,2,2,2,2)'$.

Formulation (3.1) gives the solution (1, 2, 3, 7, 14) and (1, 3, 6, 9, 13) with number of repeats 1 for each generator blocks.

Remark 3.2. In certain cases, first few (say, m) components of Λ vector may be zero. In that case, matrix **A** may be suitably generated by generating combinations such that no combination contains pairs of units separated by a distance of m units. Accordingly, it may be seen that number of such combinations ω is $\omega = \binom{v-mk-1}{k-1}$ and the dimension of the matrix **A** becomes $(d - m) \times c$. We illustrate the case by giving an example below.

Example 3.5. Consider constructing a DSBIB design with $v = 29$, $k = 4$, $\lambda_1 = 0$, $\lambda_2 = 1$ and $m = 2$. From parametric conditions in Section 4 we get $b = 58$ and $r = 8$. Here, $a_1 = m = 2$, $a_2 = d - m = 12$ and $\Lambda = (0,0,1,1,1,1,1,1,1,1,1,1,1,1)'$. Here, we generated the combinations such that minimum distance between any two elements in the combination is 3. Here, $\omega = 1140$ and $c = 165$. The formulation (3.1) gives the generator blocks (1, 4, 8, 21) and (1, 6, 12, 20) with single repeats for each. In fact, here DSBIB design reduces to a polygonal design and gives a BSA(2) plan.

Remark 3.3. Unbiased estimation of variance of Horvitz-Thompson estimator of population parameter would not be possible through such designs.

4. CONSTRUCTION AND EXISTENCE OF CYCLIC *w*-POINT DSBIB DESIGNS

First we obtain some parametric conditions for existence of cyclic w-point DSBIB designs.

Result 4.1. For odd v, if a w-point cyclic DSBIB design exists then $\frac{b}{v} = \frac{2(a_1\lambda_1 + a_2\lambda_2 + \ldots + a_w\lambda_w)}{k(k-1)}$.

Proof. Let $v = 2M + 1$. Then, maximum distance between a given unit and all other units is $d = \left[\frac{v}{2}\right] = M$. For any unit i, we get,

$$\sum_{j(\neq i)=1}^{v} \lambda_{ij} = 2(a_1\lambda_1 + a_2\lambda_2 + \ldots + a_w\lambda_w) \tag{4.1}$$

Now, for a cyclic DSBIB design,

$$\frac{b}{v} = \frac{r}{k} = \frac{\sum_{j(\neq i)=1}^{v} \lambda_{ij}}{k(k-1)} = \frac{2(a_1\lambda_1 + a_2\lambda_2 + \ldots + a_w\lambda_w)}{k(k-1)}. \tag{4.2}$$

Result 4.2. For even v, if a w-point cyclic DSBIB design exists then

$$\frac{b}{v} = \frac{2\{a_1\lambda_1 + a_2\lambda_2 + \ldots + a_{w-1}\lambda_{w-1} + (a_w - 1)\lambda_w\} + \lambda_w}{k(k-1)} \tag{4.3}$$

Corollary 4.1. For odd v, if a two-point cyclic DSBIB design exists then $\frac{b}{v} = \frac{2\{m\lambda_1 + (d-m)\lambda_2\}}{k(k-1)}$.

Corollary 4.2. For even v, if a two-point cyclic DSBIB design exists then $\frac{b}{v} = \frac{2\{m\lambda_1 + (d-m-1)\lambda_2\} + \lambda_2}{k(k-1)}$.

Corollary 4.3. For odd v, if a three-point cyclic DSBIB design exists then

$$\frac{b}{v} = \frac{2\{m_1\lambda_1 + (m_2 - m_1)\lambda_2 + (d - m_2)\lambda_3\}}{k(k-1)}.$$

Corollary 4.4. For even v, if a three-point cyclic DSBIB design exists then

$$\frac{b}{v} = \frac{2\{m_1\lambda_1 + (m_2 - m_1)\lambda_2 + (d - m_2 - 1)\lambda_3\} + \lambda_3}{k(k-1)}.$$

Now, we will consider existence of cyclic d-point DSBIB designs for $v \le 100$, $k = 3$, $\lambda_1 = 1$, $\lambda_2 = 2$, ..., $\lambda_d = d = [v/2]$.

Corollary 4.5. For odd v and $\lambda_1 = 1$, $\lambda_2 = 2$, ..., $\lambda_d = d$, if a cyclic DSBIB design exists then $v^2 \equiv 1 \mod \{4k(k-1)\}$.

Corollary 4.6. For odd v, $k = 3$ and $\lambda_1 = 1$, $\lambda_2 = 2$, ..., $\lambda_d = d$, if a cyclic DSBIB design exists then $v^2 \equiv 1 \mod (24)$.

Corollary 4.7. For even v and $\lambda_1 = 1$, $\lambda_2 = 2$, ..., $\lambda_d = d$, if a cyclic DSBIB design exists then $v^2 \equiv 0 \mod \{4k(k-1)\}$.

Corollary 4.8. For even v, $k = 3$ and $\lambda_1 = 1$, $\lambda_2 = 2$, ..., $\lambda_d = d$, if a cyclic DSBIB design exists then $v^2 \equiv 0 \mod (24)$.

We now utilize the formulation given in Section 3 to construct all existent d-point DBSPs for $N \leq 100$, $n = 3$, $\lambda_1 = 1$, $\lambda_2 = 2$, ..., $\lambda_d = d$. The parameters of the DBSPs obtained are given below:

$N \in \{5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49, 53, 55, 59, 61, 65, 67, 71, 73, 77, 79, 83, 85, 89, 91, 95, 97\}$.

In Appendix 1, we present a small catalogue of DBSPs for $n = 3$ and $N \leq 60$. The DBSPs for $N > 60$ are available with the authors and can be obtained by sending an E–mail to mandal.stat@gmail.com. We could not get any DBSP with even N for $n = 3$ in the above range using the above procedure. Therefore, this is an open problem and existence or non-existence of DBSP with even N for $n = 3$ needs to be established. We have considered the case when $\lambda_1 = 1$, $\lambda_2 = 2$, ..., $\lambda_d = d$, i.e., each successive concurrence increases by one. But this is not a restriction. Depending upon the requirements of the practical situation, one may select jumps of more than one too in some of the concurrences. The choice is dictated by the sampler but one needs to maintain $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_d$. The above procedure can be used for obtaining DBSPs for such situations as well.

5. DISCUSSION AND CONCLUSION

This article proposes a general w-point DBSP and a linear integer programming approach for obtaining cyclic w-point DBSPs for given population size, sample size and choice of distance function. The approach is very simple and effective in generating DBSPs or DSBIB designs (v, b, r, k, $\lambda_1 \mathbf{1}'_{a_1}, \lambda_2 \mathbf{1}'_{a_2}, ..., \lambda_w \mathbf{1}'_{a_w}$). Some results on existence of cyclic DSBIB designs for $k = 3$ are also obtained. Finally, we provide a list of d-point DBSPs for $n = 3$, population size $N \leq 60$, N odd and $\lambda_1 = 1$, $\lambda_2 = 2$, ..., $\lambda_d = d$.

Though the method of construction presented in the article is quite general in nature but there are some limitations of the method. Firstly, as N and n increase, the number of generator blocks increases rapidly and hence **A** matrix becomes very big which takes lot of space in computer memory and the efficiency of the algorithm reduces. Secondly, the approach presented in the article generates cyclic DBSPs. Thus, there is lot of scope for future research to develop efficient alternative methods of constructions of both cyclic and non-cyclic DBSPs.

An alternative approach to obtain DBSPs could be through using complementary property of PBIB design given by Parsad *et al*. (2007) as follows:

Result 5.1. Existence of a polygonal design {equivalent to a BSA(m) plan} with parameters v, b, r, k, λ, m implies the existence of a two-point DBSP with parameters v, b, $b - r$, $v - k$, $b - 2r$, $b - 2r + \lambda$.

Result 5.2. Existence of a w-point DBSP plan with parameters, v, b, r, k, λ_1, λ_2, ..., λ_w, implies the existence of a w-point DBSP with parameters v, b, $b - r$, $v - k$, $b - 2r$, $b - 2r + \lambda_1$, $b - 2r + \lambda_2$, ..., $b - 2r + \lambda_w$.

Using Results 5.1 and 5.2, it is possible to generate a large number of DBSPs from the existing BSA(m) plans or DBSPs. Although this approach would ensure that all the λ's are different and their values are a function of distance between the treatments (units), yet their values may be large. If one can get a BSA(m) plan with $b - 2r = 1$, then λ's would be small. Another possible way of obtaining a DBSP plan is the use of modified algorithm given by Mandal *et al*. (2008) by replacing the vector $\vartheta\mathbf{1}$ with the vector Λ for DBSP plan.

ACKNOWLEDGEMENTS

The authors would like to thank the reviewer for making valuable comments and suggestions that have led to improvements over the previous version of the paper.

REFERENCES

Hedayat, A.S., Rao, C.R. and Stufken, J. (1988). Sampling plan excluding contiguous units. *J. Statist. Plann. Inf.*, **19**, 159-170.

Horvitz, D.G. and Thompson, D.J. (1952). A generalization of sampling without replacement from a finite universe. *J. Amer. Statist. Assoc*, **47**, 663-685.

Mandal, B.N., Parsad, R. and Gupta, V.K. (2008). Computer aided construction of balanced sampling plans excluding contiguous units. *J. Statist. Appl.,* **3**(**1**), 59-85.

Mandal, B.N., Parsad, R., Gupta, V.K. and Sud, U.C. (2009). A family of distance balanced sampling plans. *J. Statist. Plann. Inf.*, **139**, 860-874.

Mandal, B.N., Gupta, V.K. and Parsad, R. (2010). Construction of polygonal designs using linear integer programming. *Comm. Statist. – Theory Methods* (To appear).

Parsad, Rajender, Kageyama, Sanpei and Gupta, V.K. (2007). Use of complementary property of block designs in PBIB designs. *Ars Combinatoria,* **85**, 173-182.

Stufken, J. (1993). Combinatorial and statistical aspects of sampling plans to avoid the selection of adjacent units. *J. Combin. Info. Syst. Sci.*, **18**, 81-92.

Stufken, J., Song, S.Y., See, K. and Driessel, K.R. (1999). Polygonal designs: Some existence and non-existence results. *J. Statist. Plann. Inf.*, **77**, 155-166.

Appendix 1. DBSP for sample size 3 for [*N*/2]-point function

N	*n*	Generator samples with number of repeats				
5	3	(1 2 4) × 1				
7	3	(1 2 5) × 1	(1 3 5) × 1			
11	3	(1 2 7) × 1	(1 3 7) × 2	(1 4 7) × 1	(1 4 8) × 1	
13	3	(1 2 8) × 1	(1 3 8) × 2	(1 4 8) × 2	(1 4 9) × 1	(1 5 9) × 1
17	3	(1 2 10) × 1	(1 3 10) × 2	(1 4 10) × 1	(1 4 11) × 2	(1 5 10) × 3
		(1 5 11) × 1	(1 6 12) × 2			
19	3	(1 2 11) × 1	(1 3 11) × 2	(1 4 11) × 2	(1 4 12) × 1	(1 5 11) × 1
		(1 5 12) × 3	(1 6 11) × 2	(1 6 12) × 1	(1 7 13) × 2	
23	3	(1 2 13) × 1	(1 3 13) × 2	(1 4 14) × 3	(1 5 13) × 4	(1 6 14) × 2
		(1 6 15) × 3	(1 7 13) × 3	(1 8 15) × 3	(1 8 16) × 1	
25	3	(1 2 14) × 1	(1 3 14) × 2	(1 4 14) × 3	(1 5 14) × 3	(1 5 15) × 1
		(1 6 14) × 2	(1 6 16) × 3	(1 7 15) × 6	(1 8 15) × 2	(1 8 17) × 3
29	3	(1 2 16) × 1	(1 3 16) × 2	(1 4 16) × 3	(1 5 16) × 1	(1 5 17) × 3
		(1 6 16) × 2	(1 6 18) × 3	(1 7 16) × 4	(1 7 17) × 2	(1 8 17) × 2
		(1 8 19) × 5	(1 9 17) × 4	(1 10 20) × 3		
31	3	(1 2 17) × 1	(1 3 17) × 2	(1 4 17) × 1	(1 4 18) × 2	(1 5 18) × 4
		(1 6 17) × 3	(1 6 19) × 2	(1 7 17) × 2	(1 7 18) × 4	(1 8 17) × 1
		(1 8 20) × 6	(1 9 17) × 4	(1 10 19) × 4	(1 11 21) × 4	
35	3	(1 2 19) × 1	(1 3 19) × 2	(1 4 19) × 1	(1 4 20) × 2	(1 5 20) × 4
		(1 6 19) × 2	(1 6 21) × 3	(1 7 20) × 6	(1 8 21) × 1	(1 8 22) × 6
		(1 9 19) × 6	(1 9 22) × 2	(1 10 19) × 4	(1 10 21) × 1	(1 11 21) × 2
		(1 12 23) × 2	(1 12 24) × 6			
37	3	(1 2 20) × 1	(1 3 20) × 2	(1 4 20) × 1	(1 4 21) × 2	(1 5 21) × 4
		(1 6 22) × 5	(1 7 20) × 5	(1 7 22) × 1	(1 8 23) × 7	(1 9 20) × 8
		(1 10 21) × 2	(1 10 24) × 7	(1 11 21) × 5	(1 12 25) × 1	(1 13 25) × 6
41	3	(1 2 22) × 1	(1 3 22) × 2	(1 4 22) × 1	(1 4 23) × 2	(1 5 23) × 4
		(1 6 22) × 4	(1 6 24) × 1	(1 7 22) × 6	(1 8 24) × 4	(1 8 25) × 3
		(1 9 23) × 8	(1 10 22) × 5	(1 10 26) × 4	(1 11 24) × 7	(1 11 25) × 3
		(1 12 23) × 1	(1 12 25) × 5	(1 12 27) × 4	(1 13 25) × 3	(1 13 27) × 1
		(1 14 28) × 1				
43	3	(1 2 23) × 1	(1 3 23) × 2	(1 4 23) × 2	(1 4 24) × 1	(1 5 24) × 4
		(1 6 24) × 3	(1 6 25) × 2	(1 7 23) × 6	(1 8 23) × 3	(1 8 25) × 4

N	n	Generator samples with number of repeats				
		(1 9 23) × 1	(1 9 26) × 7	(1 10 24) × 9	(1 11 23) × 2	(1 11 26) × 8
		(1 12 23) × 3	(1 12 28) × 5	(1 13 25) × 5	(1 14 27) × 6	(1 14 29) × 1
		(1 15 29) × 2				
47	3	(1 2 25) × 1	(1 3 25) × 2	(1 4 26) × 3	(1 5 25) × 4	(1 6 27) × 5
		(1 7 27) × 6	(1 8 25) × 5	(1 8 28) × 2	(1 9 25) × 8	(1 10 27) × 3
		(1 10 29) × 6	(1 11 26) × 4	(1 11 28) × 6	(1 12 25) × 1	(1 12 27) × 1
		(1 12 30) × 9	(1 13 25) × 1	(1 13 26) × 10	(1 14 27) × 1	(1 15 29) × 7
		(1 16 31) × 3	(1 16 32) × 4			
49	3	(1 2 26) × 1	(1 3 26) × 2	(1 4 26) × 3	(1 5 26) × 4	(1 6 27) × 4
		(1 6 28) × 1	(1 7 28) × 6	(1 8 26) × 4	(1 8 27) × 3	(1 9 26) × 4
		(1 9 27) × 4	(1 10 27) × 3	(1 10 30) × 6	(1 11 26) × 5	(1 11 28) × 5
		(1 12 27) × 3	(1 12 31) × 8	(1 13 27) × 4	(1 13 28) × 6	(1 13 29) × 2
		(1 14 30) × 8	(1 14 32) × 5	(1 15 29) × 5	(1 16 33) × 1	(1 17 33) × 3
53	3	(1 2 28) × 1	(1 3 28) × 2	(1 4 29) × 3	(1 5 29) × 4	(1 6 30) × 5
		(1 7 28) × 4	(1 7 30) × 2	(1 8 31) × 7	(1 9 28) × 8	(1 10 28) × 1
		(1 10 32) × 8	(1 11 29) × 8	(1 11 30) × 2	(1 12 28) × 5	(1 12 33) × 6
		(1 13 30) × 6	(1 13 32) × 6	(1 14 28) × 4	(1 14 34) × 9	(1 15 29) × 5
		(1 16 31) × 7	(1 16 34) × 1	(1 17 33) × 5	(1 17 34) × 1	(1 18 35) × 3
		(1 18 36) × 4				
55	3	(1 2 29) × 1	(1 3 29) × 2	(1 4 30) × 3	(1 5 30) × 4	(1 6 30) × 2
		(1 6 31) × 3	(1 7 31) × 6	(1 8 30) × 4	(1 8 32) × 3	(1 9 30) × 8
		(1 10 33) × 9	(1 11 32) × 10	(1 12 29) × 2	(1 12 34) × 9	(1 13 29) × 10
		(1 13 33) × 2	(1 14 29) × 4	(1 14 31) × 8	(1 14 35) × 1	(1 15 29) × 7
		(1 16 31) × 1	(1 16 36) × 9	(1 17 33) × 3	(1 18 35) × 1	(1 18 37) × 5
		(1 19 37) × 9				
59	3	(1 2 31) × 1	(1 3 31) × 2	(1 4 31) × 3	(1 5 32) × 4	(1 6 32) × 1
		(1 6 33) × 4	(1 7 32) × 4	(1 7 33) × 2	(1 8 31) × 6	(1 8 33) × 1
		(1 9 34) × 8	(1 10 32) × 4	(1 10 35) × 5	(1 11 31) × 10	(1 12 34) × 3
		(1 12 36) × 8	(1 13 34) × 12	(1 14 32) × 1	(1 14 36) × 8	(1 14 37) × 4
		(1 15 32) × 12	(1 15 37) × 2	(1 16 31) × 6	(1 16 33) × 1	(1 16 38) × 2
		(1 17 33) × 8	(1 18 35) × 2	(1 19 37) × 7	(1 19 38) × 1	(1 19 39) × 2
		(1 20 39) × 7	(1 20 40) × 4			