Available online at www.isas.org.in/jisas
JOURNAL OF THE INDIAN SOCIETY OF AGRICULTURAL STATISTICS 76(2) 2022 79–84

Nearly Balanced Treatment Incomplete Block Designs

B. N. Mandal, Garima Singh, Rajender Parsad and Sukanta Dash
ICAR-Indian Agricultural Statistics Research Institute, New Delhi

Received 04 June 2022; Revised 12 October 2022; Accepted 13 October 2022

SUMMARY

Balanced treatment incomplete block (BTIB) designs are quite popular for comparing test versus a single control treatment. In this article, we extend the class of BTIB designs by introducing nearly BTIB designs. Nearly BTIB designs can act as a useful alternative to BTIB designs when the latter is not available for a given parametric combination. An algorithm is proposed to construct nearly BTIB designs and a list of such designs is also provided in a practically useful parametric range.

Keywords: Nearly balanced treatment incomplete designs, BTIB, Test, Control.

1. INTRODUCTION

Experimentation is an integral part of a research investigation. The designs used for experimentation mostly focus on all possible paired comparisons among treatments. But there are situations where interest of the experimenter is only in some subset of all possible paired comparisons. The experimenter may be interested to compare a set of new treatments called test treatments with one established treatment called control. Often there may be a nuisance factor that needs to be taken care of during experimentation. Block designs are used in that situation. To make the exposition clearer, let the control be labelled as 1 and v test treatments be labelled as 2, 3, ..., $v + 1$ and these are to be compared with the control. Consider the additive fixed effect linear model

$$y_{iju} = \mu + \tau_i + \beta_j + \varepsilon_{iju} \tag{1}$$

where y_{iju} denotes the observation from experimental unit u in block j receiving ith treatment, μ denotes the general mean, τ_i denotes the effect of ith treatment, β_j denotes the effect of jth block and ε_{iju} are random uncorrelated errors with mean 0 and constant variance σ^2; $i = 1, 2, ..., v + 1$; $j = 1, 2, ..., b$; $u = 1, 2, ..., k$. The contrasts of interest for comparing test treatments versus the control are $\tau_i - \tau_1$, $i = 2, 3, ..., v + 1$.

A very important class of block designs namely balanced treatment incomplete block (BTIB) designs (Bechhofer and Turnbull, 1971) is available in literature for comparing test versus control problem. A number of work is available on obtaining BTIB designs and A-optimal BTIB designs for comparing test treatments with a single control, see for example, Hedayat and Majumdar (1984, 1985); Stufken (1987, 1988); Cheng *et al.* (1988); Hedayat *et al.* (1988); Jacroux (1989); Gupta (1989); Das *et al.* (2005); Mandal *et al.* (2017) and Mandal *et al.* (2020). However, there may be many parametric combinations for which a BTIB design does not exist. Sometimes for given parameters, even if a BTIB design exists, it may require large number of replications of the control. For example, there does not exist a BTIB design for $v = 5$, $b = 10$, $k = 4$ with less than 10 replication of the control. A non-BTIB design with these parameters is given in Table 1 where the control has only 5 replications. Thus, in such situations there is a need to look beyond the class of BTIB designs for comparing test treatments with a control. One solution is to develop a class of designs which retains the most of the characteristics of BTIB designs. In this article,

Corresponding author: B.N. Mandal
E-mail address: mandal.stat@gmail.com

we introduce one new class of such designs called nearly BTIB designs for comparing test treatments with a single control treatment.

Table 1. A Non-BTIB design for $v = 5$, $b = 10$, $k = 4$ with $r_1 = 5$

(1	4	5	6)
(1	3	5	6)
(2	3	5	6)
(3	4	5	6)
(2	4	5	6)
(1	2	3	4)
(1	2	4	6)
(1	2	3	5)
(2	3	4	6)
(2	3	4	5)

The article is organized as follows. In Section 2, we give the definition of nearly BTIB designs. An algorithm is presented to construct nearly BTIB designs in Section 3. Working of the algorithm is illustrated in Section 4. A list of nearly BTIB designs constructed using the proposed algorithm is given in Section 5. Concluding remarks are given in Section 6.

2. NEARLY BALANCED TREATMENT INCOMPLETE BLOCK DESIGN

Let $\mathbf{N} = (n_{ij})$ denotes $(v + 1) \times b$ treatment versus block incidence matrix where n_{ij} denotes the number of times ith treatment ($i = 1, 2, ..., v + 1$) appears in jth block ($j = 1, 2, ..., b$). We define nearly balanced treatment incomplete block designs below.

Definition 1. A block design is said to be nearly balanced treatment incomplete block (nearly BTIB) design if the following conditions hold:

$$\sum_{j=1}^{b} n_{1j} = r_1$$

$$\sum_{j=1}^{b} n_{ij} = r_2, i = 2, 3, ..., v+1$$

$$\sum_{j=1}^{b} n_{1j} n_{ij} = \lambda_1, i = 2, 3, ..., v+1 \text{ and}$$

$$\sum_{j=1}^{b} n_{ij} n_{i'j} = \lambda_2 \text{ or } \lambda_2 + 1, i \neq i' = 2, 3, ..., v+1$$

In other words, in a nearly BTIB design, the concurrence of each test treatment with the control is λ_1 and the concurrence of a test treatment with any other test treatment is λ_2 or $\lambda_2 + 1$. The statistical implication of this is that (a) test treatments are compared with the control with equal precision and (b) test treatments among themselves are not compared with equal precision and clearly, interest of the experimenter is protected with (a).

Example 1. A nearly BTIB design for $v = 5$, $b = 5$, $k = 4$, $r_1 = 5$, $r_2 = 3$, $\lambda_1 = 3$, $\lambda_2 = 1$ is given in Table 2.

Table 2. A nearly BTIB design with $v = 5$, $b = 5$, $k = 4$

(1 4 5 6)
(1 3 4 6)
(1 2 5 6)
(1 2 3 5)
(1 2 3 4)

From the above definition, it is clear that a nearly BTIB design may be binary or nonbinary in test and control treatments. However, for the sake of simplicity, we further assume in this article that the design is binary in test and control treatments. We further assume that each test treatment appears with n_1 test treatments λ_2 times and with n_2 test treatments $\lambda_2 + 1$ times with $n_1 + n_2 = v - 1$. In that situation, it follows that each of the test treatments has equal number of replications, denoted as r_2. Let the number of replications of the control be r_1. Hence, the parameters of a nearly BTIB design are v, b, k, r_1, r_2, λ_1, λ_2, n_1, n_2. Parametric relations among these parameters are given below.

$$\begin{aligned} & r_1 + v r_2 = bk \\ & n_1\lambda_2 + n_2(\lambda_2 + 1) + \lambda_1 = r_2(k-1) \\ & \text{such that } n_1 + n_2 = v - 1 \\ & r_1(k-1) = v\lambda_1 \end{aligned} \quad (2)$$

The first parametric relation follows from the fact that total number of units required with r_1 replications of the control and r_2 replications of v test treatments is $r_1 + vr_2$. Also, the total number of units from b blocks each of size k is bk. The second relation can be proved as follows. Consider any test treatment. Since it appears with n_1 other test treatments in λ_2 blocks and with n_2 other test treatments in $\lambda_2 + 1$ blocks and with the control treatment in λ_1 blocks, hence total number of pairs with this test treatment is $n_1\lambda_2 + n_2(\lambda_2 + 1) + \lambda_1$. But this test treatment appears in r_2 blocks and in those blocks there are $(k - 1)$ other positions. Hence, total number of pairs with this test treatment in these

r_2 blocks is $r_2(k-1)$, hence the equality. Next relation can be proved similarly. It may be mentioned here that the parametric conditions in (2) are necessary and not sufficient for the existence of a nearly BTIB design. The structure of the concurrence matrix of a nearly BTIB design is thus

$$\mathbf{NN}' = \left(\begin{array}{c|cccc} r_1 & \lambda_1 & \lambda_1 & \cdots & \lambda_1 \\ \hline \lambda_1 & r_2 & \lambda_2 & \cdots & \lambda_2+1 \\ \lambda_1 & \lambda_2 & r_2 & \cdots & \lambda_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_1 & \lambda_2+1 & \lambda_2 & \cdots & r_2 \end{array}\right) \quad (3)$$

Clearly, the concurrence matrix is of order $(v+1) \times (v+1)$. In each of the rows, from the 2nd row to $(v+1)$th row, there are n_1 positions with λ_2 and n_2 positions with $\lambda_2 + 1$. The same is true for 2nd column to $(v+1)$ th column also. However, varying positions of λ_2 and of $(\lambda_2 + 1)$ in the concurrence matrix may lead to different efficiencies of the designs. We do not study this aspect in this article and may be explored while searching for 'efficient' nearly BTIB designs for given parameters in a future article.

3. ALGORITHM

In this Section, we propose an algorithm to construct nearly BTIB designs for specified parameters. This algorithm is a modification of the algorithm of Mandal *et al.* (2017) to construct BTIB designs. The steps of the algorithm are described in detail below.

Step 1: For given v, b, k, λ_1, find r_1, r_2, λ_2, n_1, n_2 by using the necessary conditions given in (2). If all of these are integers, then proceed, otherwise a nearly BTIB design does not exist for given v, b, k, λ_1.

Step 2: Obtain a concurrence matrix $\mathbf{NN}'$ as in (3) with the properties as described above. Denote this matrix as $\mathbf{\Lambda}$ for easy reference.

Step 3: In this step, the algorithm tries to obtain an incidence matrix $\mathbf{N}$ so that this incidence matrix gives the concurrence matrix obtained in Step 2. The rows of the incidence matrix are obtained one by one. The first row is obtained at random such that randomly chosen r_1 positions of the row are filled with 1 and remaining $b - r_1$ positions of the row are filled with 0. Denote this $1 \times b$ matrix as $\mathbf{N}^{(1)}$. Set another matrix $\mathbf{T} = \Phi$ where Φ is a null matrix. Since ith ($i = 2, 3, ..., v+1$) row of $\mathbf{N}$ is unknown, let the elements of ith row be $(x_1, x_2, ..., x_b)$. To obtain ith row, first obtain weights $w_j = 1/k_j$ if $k_j > 0$ and $w_j = 1$ if $k_j = 0$ where $k_j = \sum_{i'=1}^{i-1} n_{i'j}$, $j = 1, 2, ..., b$ with $n_{i'j}$ being the element at i' th row and jth column of $\mathbf{N}^{(i-1)}$, $i' = 1, 2, ..., i-1$; $j = 1, 2, ..., b$. Now, solve the following linear integer programming problem for the row i with respect to binary decision variables x_1, x_2, ..., x_b :

$$\text{Maximize } \varphi = \sum_{j=1}^{b} w_j x_j \text{ subject to constraints}$$
$$\begin{aligned} &\sum_{j=1}^{b} x_j = \Lambda_{ii} \\ &x_j \le k - k_j,\ j = 1,2,...,b \\ &\sum_{j=1}^{b} n_{i'j} x_j = \Lambda_{i'i},\ i' = 1,2,...,\ i-1 \end{aligned} \quad (4)$$

where Λ_{ab} denotes the (a, b)th element of $\mathbf{\Lambda}$. If there exists an optimal solution of the formulation (4), then set $\mathbf{N}^{(i)} = \begin{pmatrix} \mathbf{N}^{(i-1)} \\ \mathbf{x}_0' \end{pmatrix}$ where $\mathbf{x}_0' = (x_{01}, x_{02}, ..., x_{0b})'$ denotes an optimal solution to the formulation (4) and then go to next i.

If there is no optimal solution of (4), select a random number m between 2 to $(i-1)$, set $\mathbf{T} = \begin{pmatrix} \mathbf{T} \\ \mathbf{n}_m^{(i-1)} \end{pmatrix}$ where $\mathbf{n}_m^{(i-1)}$ denotes the mth row of the $\mathbf{N}^{(i-1)}$ matrix, and then set $\mathbf{n}_m^{(i-1)} = \mathbf{0}'$ in the $\mathbf{N}^{(i-1)}$ matrix. To obtain row m, solve the formulation (5) with respect to binary decision variables $x_1, x_2, ..., x_b$:

$$\text{Maximize } \varphi = \sum_{j=1}^{b} w_j x_j \text{ subject to constraints}$$
$$\begin{aligned} &\sum_{j=1}^{b} x_j = \Lambda_{ii} \\ &x_j \le k - k_j,\ j = 1,2,...,b \\ &\sum_{j=1}^{b} n_{i'j} x_j = \Lambda_{i'm},\ i' = 2,...,m-1,m+1,...,i-1 \\ &\sum_{j=1}^{b} t_{qj} x_j < r_2,\ q = 1,2,...,p \end{aligned} \quad (5)$$

where p is the number of rows of the matrix $\mathbf{T}$, and t_{qj} is the element at the qth row and the jth column ($q = 1, 2, ..., p, j = 1, 2, ..., b$) of the $\mathbf{T}$ matrix. If there is an optimal solution to (5), then update $\mathbf{N}^{(i-1)}$ by setting $\mathbf{n}_m^{(i-1)} = \mathbf{x}_0'$. If there is no solution to (5), repeat this step by drawing another random number m. If the deleted row of $\mathbf{N}^{(i-1)}$ is not obtained after a certain number say,

n_0, of times, then start afresh with step 3 with $i = 2$. Starting afresh will be counted as one trial. Algorithm will terminate if no solution is obtained after number of trials reaches a threshold preset at say, 100.

If all $v + 1$ rows of the incidence matrix is obtained, then the $\mathbf{N}^{(v+1)}$ matrix is a desired incidence matrix of a nearly BTIB design. Calculate A-efficiency $e_A(d)$ of this design using the following formula:

$$e_A(d) = \frac{g(t,s)}{Trace(\mathbf{PC}_d^{-}\mathbf{P}')} \tag{6}$$

where $g(t, s)$ is the minimum value of $g(x, z)$ defined in equation (2.5) of Hedayat and Majumdar (1984), $\mathbf{P} = (\mathbf{1}_v \vdots -\mathbf{I}_v)$ with $\mathbf{I}_v$ denoting an identity matrix of order v and $\mathbf{1}_v$ denoting a $v \times 1$ column vector of 1's and $\mathbf{C}_d = \mathbf{R}_d - \frac{1}{k}\mathbf{N}_d\mathbf{N}'_d$ with $\mathbf{R}_d$ being the diagonal matrix of number of replications of treatments in design d. The output is, thus, the required design along with its A-efficiency.

Remark 1. The formulation (4) allocates treatments to the ith row of the incidence matrix **N** with $x_j = 1$ means jth block is allocated treatment i. The objective function maximizes allocation of treatment i to those blocks where least number of allocations has been done till $(i - 1)$ rows of **N**. The first constraint ensures treatment i is allocated to as many blocks as equal to its number of replications. The second set of constraints ensure block sizes do not exceed k. The third set of constraints ensure that the concurrences of the ith treatment with treatments 1 to $i - 1$ are achieved. The formulation (5) is to obtain another solution for deleted row m, $m \in \{2, 3, ...,(i - 1)\}$ of **N** with similar interpretations for constraints. The additional fourth constraint in formulation (5) does not allow already deleted row to be a solution.

Remark 2. The algorithm terminates in two conditions: firstly, when a nearly BTIB design is obtained or secondly when the number of trials exceeds the preset threshold number of trials. It may happen that even if a nearly BTIB design exists, proposed algorithm may not give us a design. Proposed algorithm can be used to generate nearly BTIB designs for $v \leq 30$ and $k \leq 10$. For larger values of v, the chances of entering improper candidate rows in **N** matrix increases in the intermediate steps which leads to no solution in the subsequent steps and ultimately complete **N** is not obtained. Identifying culprit rows could be a game changer. However, we not have any clue on this till now and it is an open problem.

4. WORKING OF THE ALGORITHM

In this Section, we illustrate the working of the algorithm with the help of an example. Consider construction of a nearly BTIB design with parameters $v = 4$, $b = 6$, $k = 3$, $\lambda_1 = 1$. Step 1 of the algorithm gives $r_1 = 2$, $r_2 = 4$, $n_1 = 2$, $n_2 = 1$ from necessary conditions. Step 2 of the algorithm gives a concurrence matrix with properties as in (3) as given below

$$\mathbf{NN}' = \begin{pmatrix} 2 & 1 & 1 & 1 & 1 \\ 1 & 4 & 3 & 2 & 2 \\ 1 & 3 & 4 & 2 & 2 \\ 1 & 2 & 2 & 4 & 3 \\ 1 & 2 & 2 & 3 & 4 \end{pmatrix}$$

In Step 3, first $\mathbf{N}^{(1)}$ is obtained as $\mathbf{N}^{(1)} = (1\ 0\ 1\ 0\ 0\ 0)$. To obtain row 2 of the incidence matrix, the algorithm computes weights $w_j = 1$, $j = 1, 2, ..., 6$ and solves the linear integer programming problem

Maximize $\varphi = \sum_{j=1}^{6} x_j$ subject to constraints

$$\sum_{j=1}^{6} x_j = 4$$

$x_1 \leq 2$

$x_2 \leq 3$

$x_3 \leq 2$

$x_4 \leq 3$

$x_5 \leq 3$

$x_6 \leq 3$

$x_1 + x_3 = \lambda_1 = 1$

An optimal solution to this formulation and hence, the second row of **N** matrix is $(0\ 0\ 1\ 1\ 1\ 1)$. Thus, $\mathbf{N}^{(2)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$. For $i = 3$, the weights are obtained as $w_j = 0.5$, $\forall\, j = 3$, and $w_j = 1$ for all other j's. Hence, the formulation for $i = 3$ is the following

Maximize $\phi = x_1 + 0.5x_3 + x_4 + x_5 + x_6$ subject to constraints

$$\sum_{j=1}^{6} x_j = 4$$

$x_1 \leq 2$

$x_2 \leq 3$

$x_3 \leq 1$

$x_4 \leq 2$

$x_5 \leq 2$

$x_6 \leq 2$

$x_1 + x_3 = \lambda_1 = 1$

$x_3 + x_4 + x_5 + x_6 = 2$

An optimal solution to this formulation is (0 1 1 0 1 1) and hence $\mathbf{N}^{(3)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$. This process is continued for $i = 4, 5$, and at the end of Step 6, we get the following **N** matrix

$$\mathbf{N}^{(5)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

This is an incidence matrix of a nearly BTIB design with the given parameters and the block contents are

(1 4 5)

(3 4 5)

(1 2 3)

(2 4 5)

(2 3 5)

(2 3 4)

5. LIST OF NEARLY BTIB DESIGNS

In this Section, we provide a list of nearly BTIB designs constructed using the proposed algorithm in Section 3. Though the algorithm proposed is general in nature, however, we have used the algorithm to construct designs in a restricted parametric range $v \leq 30$, $b \leq 40$, $2 \leq k \leq \min(v, 10)$, $\lambda_1 \leq 5$. We denote this restricted parametric range as P. In the above parametric range, a total of 635 sets of parameters satisfy necessary conditions in (2). Block size wise distribution of these 635 designs along with number of designs obtained through the proposed algorithm and number of designs for which solution could not be obtained is given in Table 3.

Table 3. Nearly BTIB designs in P

	k									**Total**
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	
Number of designs	60	111	96	89	103	63	32	50	31	635
Number of designs obtained	30	111	95	69	93	59	32	47	29	595
Number of designs for which solution is unknown	0	0	1	20	10	4	0	3	2	40

Distribution of A-efficiencies of obtained designs is given in Table 4.

Table 4. Distribution of A-efficiencies of nearly BTIB designs

A-efficiency	**≤ 0.5**	**0.5-0.6**	**0.6-0.7**	**0.7-0.8**	**0.8-0.9**	**0.9-1.00**	**Total**
Number of designs	0	9	218	283	85	0	595

The list of 595 nearly BTIB designs along their layouts is available at the webpage https://drs.icar.gov.in/nbtib/NBTIB.htm (Singh *et al.* 2017).

Nearly BTIB designs which were obtained in P were also compared with BTIB designs available in Mandal *et al.* (2013). There are 1361 A-efficient BTIB designs listed by Mandal *et al.* (2013). There are 314 nearly BTIB designs and BTIB designs with same v, b, k. Out of these 314 designs, there are 182 nearly BTIB designs which are more A-efficient than corresponding BTIB designs of Mandal *et al.* (2013).

Remark 3. Experiments for test versus control comparisons are often conducted in many research investigations and for this purpose, designs for test versus control comparison are useful. List of nearly BTIB designs presented above will be further addition in the kitty of available designs such as BTIB designs for such experiments and will be useful to researchers especially agricultural experimenters who require designs for test versus control comparisons. Depending on the requirement of an experiment with regards to number of treatments, number of blocks, block size, number of replications of the test treatments and control, experimenters can suitably choose a BTIB or a nearly BTIB design.

6. CONCLUDING REMARKS

We have introduced a new class of designs called nearly balanced treatment incomplete block designs for comparing test treatments with a single control. These designs will enrich the class of block designs for test versus control comparisons. We have proposed an algorithm to construct such designs and also obtained a number of such designs in a practically useful parametric range. It is natural to believe that most of these are new. Further research efforts may be directed towards characterization of these designs and development of algebraic methods of constructions.

ACKNOWLEDGEMENTS

This article is a part of M.Sc. thesis work by Ms. Garima Singh (2018) carried out in ICAR-Indian Agricultural Research Institute, New Delhi. The authors are thankful to the reviewer for valuable suggestions which led to the improvement in the presentation of the article.

REFERENCES

Bechhofer, R.E. and Turnbull, B.W. (1971). Optimal allocation of observations when comparing several treatments with a control. iii: Globally best one-sided intervals for unequal variances. Technical report, DTIC Document.

Cheng, C.S., Majumdar, D., Stufken, J., and T¨ure, T.E. (1988). Optimal step-type designs for comparing test treatments with a control. *Journal of the American Statistical Association*, **83(402)**, 477-482.

Das, A., Dey, A., Kageyama, S., and Sinha, K. (2005). A efficient balanced treatment incomplete block designs. *Australasian Journal of Combinatorics*, **32**, 243-252.

Gupta, S. (1989). Efficient designs for comparing test treatments with a control. *Biometrika*, **76(4)**, 783-787.

Hedayat, A., Jacroux, M., and Majumdar, D. (1988). Optimal designs for comparing test treatments with controls. *Statistical Science*, **3(4)**, 462-476.

Hedayat, A. and Majumdar, D. (1984). A-optimal incomplete block designs for controltest treatment comparisons. *Technometrics*, **26(4)**, 363-370.

Hedayat, A. and Majumdar, D. (1985). Families of A-optimal block designs for comparing test treatments with a control. *The Annals of Statistics*, **13(2)**, 757-767.

Jacroux, M. (1989). The A-optimality of block designs for comparing test treatments with a control. *Journal of the American Statistical Association*, **84(405)**, 310-317.

Mandal, B.N., Gupta, V. K., and Parsad, R. (2017). Balanced treatment incomplete block designs through integer programming. *Communications in Statistics - Theory and Methods*, **46(8)**, 3728-3737.

Mandal, B.N., Parsad, R., and Dash, S. (2020). Construction of A-optimal balanced treatment incomplete block designs: An algorithmic approach. *Communications in Statistics: Simulation and Computation*, **49(6)**, 1653-1664.

Mandal, B.N., Parsad, R., and Gupta, V.K. (2013). Balanced treatment incomplete block designs for test treatments vs control comparisons: Design resources server. https://drs.icar.gov.in/btib/btib.htm.

Singh, G., Mandal, B.N., Parsad, R., and Dash, S. (2017). Nearly balanced treatment incomplete block designs. http://drs.icar.gov.in/nbtib/NBTIB.htm.

Stufken, J. (1987). A-optimal block designs for comparing test treatments with a control. *The Annals of Statistics*, **15(4)**, 1629-1638.

Stufken, J. (1988). On bounds for the efficiency of block designs for comparing test treatments with a control. *Journal of Statistical Planning and Inference*, **19(3)**, 361-372.