TECHNOMETRICS, AUGUST 1994, VOL. 36, NO. 3

Construction of Optimal Block Designs by Computer

Nam-Ky NGUYEN

CSIRO-IAPP Biometrics Unit
Private Bag 10, Rosebank MDC
Clayton, Victoria 3169
Australia

This article describes an effective algorithm for constructing optimal or near-optimal incomplete block designs with up to 100 treatments. The algorithm is found to perform well when evaluated against 874 optimal or near-optimal incomplete block designs in the literature. Examples that motivated the development of the algorithm are given.

KEY WORDS: α designs; A optimality; Computer-aided designs; DETMAX; Design augmentation; Generalized cyclic designs; Incomplete block designs; Interchange algorithm; Regular graph designs.

In many experimental situations, the number of treatments is so large that it is not possible to accommodate a replicate of every treatment in each block. It is therefore necessary to use designs in which the number of treatments is larger than the block size. We call these designs *incomplete block designs* (IBD's). Although IBD's have a history associated with agricultural experiments carried out at Rothamsted under the direction of Sir Ronald Fisher and Frank Yates in the 1930s, they quickly became of great value in many other areas of experimental science.

Over the years, existing catalogs of IBD's such as that of Clatworthy (1973) have been used to obtain satisfactory solutions for many experimental situations. It is not rare, however, for an experimenter to experience situations in which they fail spectacularly. The following are a few examples of the latter:

1. The removal of lichens such as *Lecanora dispersa* from asbestos cement (AC) roofing by mechanical methods—for examle, by high-pressure water jet—poses health and safety problems because of the release of large amounts of asbestos fibers. The alternative is some form of chemical treatment including the use of commercial fungicides, disinfectants, and quaternary ammonium compounds. Therefore, an experiment was conducted to compare the efficacies of 14 chemical treatments in preventing the lichen growth on AC roofing. Twenty-eight corrugated AC panels were available for this experiment. Because the resistance of lichens to chemical treatments was expected to vary from panel to panel, the panels were treated as blocks. It was feasible to divide each panel into five strips and to treat each strip with a different chemical treatment. Thus each panel received an incomplete set of treatments and each treatment appeared 10 times. A plan of a partially balanced IBD (PBIBD) for 14 treatments each with 10 replications in 28 blocks of size 5 was not available from Clatworthy (1973), but the one for 15 treatments was (plan **R150**). It was not possible, however, to frame an additional treatment to allow the use of this plan.

2. Fifteen brands of paint were tested for weathering stability under ultraviolet (UV) irradiation. The UV weatherometer could only hold three paint panels at a time. Ten runs (blocks) were made. The design was an IBD for 15 treatments each with two replications in blocks of size 3. After the analysis of the data, it was realized that the expected level of precision of the results could only be achieved by having at least an additional replication of the treatments. Existing literature on IBD's does not provide any clue as to how this can be carried out.

3. An in-ground natural durability field test is to be conducted to study the service life of 21 commercial timber species including western red cedar and California redwood. Each species will be replicated 10 times in 35 blocks of size 6. For ease of management, the 35 blocks were divided into five subsets. Each subset is an IBD for 21 treatments each with two replications in seven blocks of size 6. Designs of this type are called *two-resolvable* designs. A candidate two-resolvable design for this test was the PBIBD in plan **T69** of Clatworthy (1973). Common sense indicates, however, that this is a very poor design because it does not make sense for certain pairs of treatments to appear together in five blocks while certain pairs never appear together in any of the blocks.

Clearly, in the days of widespread computer availability there is a need for developing good computer algorithms that will enable experimenters to construct designs in a broader range of experimental situations and to replace catalogs with more user-friendly computer programs. This work, however, has not been accomplished by pioneering computer-aided design works of Mitchell (1973), Jones (1976), Jones and Eccleston (1980), Russell, Eccleston, and Knudsen (1981) or recent works of Whitaker, Triggs, and John (1990) and Nguyen and Dey (1990). The computer algorithms described by these authors can only be used effectively to construct fairly small designs with no more than 12 treatments. The only outstanding exception is the α algorithm (Paterson and Patterson 1983; Patterson and Williams 1976a), which restricts the search to designs of the cyclic family. The purpose of this article is to present an effective general algorithm for constructing optimal or near-optimal IBD's with up to 100 treatments and to evaluate its performance against 874 optimal or near-optimal IBD's in the literature.

1. A CRITERION FOR COMPARING DESIGNS

An IBD of size (v, k, r) has v treatments set out in b blocks of size k $(<v)$ such that each treatment is replicated r times. Assume that no treatment occurs more than once in a block. An IBD is said to be α *resolvable* if the blocks can be divided into subsets, each of which is an IBD of size (v, k, α). A one-resolvable IBD is a resolvable IBD. The information matrix for the adjusted treatment effects of an IBD of size (v, k, r) is

$$C_{v\times v} = rI - k^{-1}NN'. \tag{1}$$

$NN' = \{\lambda_{tt'}\}$ is the treatment *concurrence* matrix in which $\lambda_{tt} = r(t = 1, \ldots, v)$ and $\lambda_{tt'}(t \neq t')$ is the number of blocks in which treatments t and t' both appear. When $\lambda_{tt'}$'s $= \lambda(t \neq t')$, the design is called a *balanced* IBD (BIBD). When the $\lambda_{tt'}$'s $(t \neq t')$ differ by at most 1, the design is called a regular graph design (RGD) (John and Mitchell 1977; Mitchell and John 1976).

A common criterion for experimenters to compare designs of the same size is the efficiency factor defined as $E = (v - 1)/\Sigma\, e_t^{-1}$, where $e_t(t = 1, \ldots, v - 1)$ are the nonzero roots of $r^{-1}C$ assuming that the design is *connected*; that is, $\text{rank}(C) = v - 1$. A design that has the maximal value of E is said to be A-optimal (John 1987, sec. 2.4; Kiefer 1959). A rationale for the preference of A optimality is that it minimizes the average standard error of treatment difference estimates under the usual distribution assumptions.

Algorithms working directly with A optimality—that is, maximizing E or minimizing $\Sigma\, e_t^{-1}$ (e.g., Jones and Eccleston 1980; Venables and Eccleston 1993)—are computationally intensive because they need to work with C^-, the generalized inverse of C. Another approach is to work with an approximation to A optimality. Note that the expansion of the generalized inverse of $r^{-1}C$ in (1) gives

$$\sum e_t^{-1} = \text{constant} + \sum_{i=2}^{\infty} (rk)^{-i}\, \text{tr}(NN')^i \tag{2}$$

(eq. 2.14 of John 1987). Thus minimizing the whole series in the right side of (2) is the same as minimizing $\Sigma\, e_t^{-1}$. It can be seen that the minimum of (2) is attained with a BIBD. As the traces of the higher powers of NN' contribute least to $\Sigma\, e_t^{-1}$, it is sensible to minimize successively the first few terms of this series. This is the strategy adopted by Hall and Jarrett (1981), Lamacraft and Hall (1982), and Paterson (1983) for the construction of nonresolvable generalized cyclic designs (GCD's) and resolvable GCD's (α designs) and is also the strategy used in this article. Specifically, this strategy involves two stages:

1. Minimize the first term in the series. This is equivalent to minimizing

$$f_2 = \sum_{t=1}^{v-1} \sum_{t'=t+1}^{v} \lambda_{tt'}^2. \tag{3}$$

Designs that minimize f_2 are called (M, S)-optimal (Eccleston and Hedayat 1974; Shah 1960). The ultimate aim of this stage is to find an (M, S)-optimal design that is either a BIBD or an RGD with two concurrences λ and $\lambda + 1$. If such an RGD is found, proceed to stage 2.

2. Keep f_2 constant and minimize the second term in this series. This is equivalent to minimizing

$$f_3 = \sum_{t=1}^{v-2} \sum_{t'=t+1}^{v-1} \sum_{t''=t'+1}^{v} \lambda_{tt'}\lambda_{tt''}\lambda_{t't''} \tag{4}$$

while keeping the RGD status of the design unaltered.

Lower bounds for f_2 and f_3 in (3) and (4) can be derived from section 2.1 of Jarrett (1989). Note that in stage 1 the lower bound of f_2 is reached when the design becomes a BIBD or an RGD with two concurrences λ and $\lambda + 1$.

2. THE ALGORITHM

My algorithm for constructing IBD's is an *iterative improvement* algorithm belonging to the class of *interchange* algorithms advocated by Jones (1976) and subsequently followed by Jones and Eccleston (1980) and Russell et al. (1981). In these algorithms, each iteration involves a swap of two treatments from two different blocks. There is an analogy between these algorithms and the edge-exchange algorithm used in

the traveling-salesman problem (Lin and Kernighan 1973). Of the mentioned algorithms, that of Jones and Eccleston (1980) is the most known and successful. This algorithm is an improvement on the algorithm of Jones (1976) in two ways:

1. An *exchange* stage in the essence of the DETMAX exchange algorithm (Mitchell 1973, 1974; Nguyen and Miller 1992) is carried out prior to the *interchange* stage to speed up the convergence of the starting design to the optimal or near-optimal design.
2. A set of recursive formulas (eqs. 2.1–2.4 of Jones and Eccleston 1980) is used to facilitate the update of C^-.

My algorithm is also an attempt to improve the algorithm of Jones (1976) but in a different direction:

1. I repeat my algorithm several times called *tries*. This repetition is the mechanism to evade local optima in an iterative improvement algorithm (Lin and Kernighan 1973; Mitchell 1973, 1974; Nguyen and Miller 1992);
2. To afford several tries without a substantial increase in computer time, I use f_2 and f_3 instead of E as my objective functions. The use of these objective functions facilitates the speedy calculation of the *delta functions*, which I shall describe later.
3. Moreover, to ensure that no try is wasted, I implement a *stopping* rule for each (v, k, r) combination.

Now, let t^{ij} denote a treatment in the jth position of block i and let t^{mn} denote a treatment in the nth position of block m of an IBD of size (v, k, r) (t^{ij} is not in block m and t^{mn} is not in block i). We want to know the effect of swapping t^{ij} and t^{mn} on f_2 and f_3. Let S_i denote the set of treatments in block i other than t^{ij} and other than those in block m. Similarly, let S_m denote the set of treatments in block m other than t^{mn} and other than those in block i.

Let $t \in S_i$. The pairwise swap of t^{ij} and t^{mn} will increase all $\lambda_{tt^{mn}}$'s by 1 and decrease all $\lambda_{tt^{ij}}$'s by 1. The concurrences associated with the treatments in S_m and t^{ij} (or t^{mn}) will also change accordingly. Hence it can be seen that the swap of t^{ij} and t^{mn} will increase f_2 in (3) by an amount

$$\Delta f_2(t^{ij}, t^{mn}) = 2\left\{\sum_{t\in S_i}(\lambda_{tt^{mn}} - \lambda_{tt^{ij}} + 1) + \sum_{t\in S_m}(\lambda_{tt^{ij}} - \lambda_{tt^{mn}} + 1)\right\} \quad (5)$$

by noting that only some squared terms in the right side of (3) of the form $\lambda_{tt^{ij}}$ and $\lambda_{tt^{mn}}$ might change value after the swap.

Now, if the design is an RGD with two concurrences 0 and 1 and if this status is retained after the swap—that is, $\Delta f_2(t^{ij}, t^{mn}) = 0$—the swap of t^{ij} and t^{mn} will increase f_3 in (4) by an amount

$$\Delta f_3(t^{ij}, t^{mn}) = \sum_{t=1}^{v-1}\sum_{t'=t+1}^{v} g(t, t'), \quad (6)$$

where $g(t, t') = 0$ if $\lambda_{tt'} = 0$ or $t \in \{t^{ij}, t^{mn}\}$ or $t' \in \{t^{ij}, t^{mn}\}$. And $g(t, t')$

$= \lambda_{t't^{ij}} - \lambda_{t't^{mn}}$ if $t \in S_m$ and $t' \notin S_i$ and $t' \notin S_m$

$= \lambda_{t't^{mn}} - \lambda_{t't^{ij}}$ if $t \in S_i$ and $t' \notin S_i$ and $t' \notin S_m$

$= \lambda_{tt^{ij}} - \lambda_{tt^{mn}}$ if $t' \in S_m$ and $t \notin S_i$ and $t \notin S_m$

$= \lambda_{tt^{mn}} - \lambda_{tt^{ij}}$ if $t' \in S_i$ and $t \notin S_i$ and $t \notin S_m$

$= 0$ otherwise.

by noting that only some triplets in the right side of (4) of the form $\lambda_{tt'}\lambda_{tt^{ij}}\lambda_{t't^{ij}}$ and $\lambda_{tt'}\lambda_{tt^{mn}}\lambda_{t't^{mn}}$ might change value after the swap. For RGD's with two concurrences $\lambda(\neq 0)$ and $\lambda + 1$, I transform NN' to $NN' - \lambda J$ before minimizing f_3. This is feasible because NN' has a constant row sum, so minimizing $\mathrm{tr}(NN')^3$ is the same as minmizing $\mathrm{tr}(NN' - \lambda J)^3$. We call Δf_2 in (5) and Δf_3 in (6) *delta functions* of f_2 and f_3. Since I only work with the $\frac{1}{2}v(v-1)$ elements above the diagonal elements of NN', in this article $\lambda_{tt'}$ corresponds to the element in the tth row and t'th column of this matrix if $t < t'$ and to the element in the t'th row and tth column if $t > t'$.

The steps of my algorithm for each try using the derived delta functions for f_2 and f_3 are as follows:

1. Construct a *starting* design D by allocating the 1st replicate of v treatments at random to the blocks. Continue with the 2nd, . . . , rth replicate until all the blocks are filled up. The only constraint in this allocation is that no treatment is allowed to occur in a block more than once. Update NN'.
2. Calculate f_2. If f_2 reaches its lower bound, go to step 3; otherwise proceed as follows. For a pair of blocks i and m among $\binom{b}{2}$ possible blocks, use (5) to repeat searching for a pair of treatments t^{ij} in block i and t^{mn} in block m (t^{ij} is not in block m and t^{mn} is not in block i) such that the swap of these two treatments results in the biggest reduction in f_2. If the search is successful, update f_2, NN' and D. If f_2 cannot be reduced further, go to the next pair of blocks. This process is repeated until f_2 cannot be reduced further or there is no further gain by any further possible treatment swap. If f_2 reaches its lower bound, go to step 3.
3. If the current design is an RGD with two concurrences λ and $\lambda + 1$ (i.e., not a BIBD), calculate f_3. If f_3 reaches its lower bound, skip this step; other-

wise proceed as follows. For a pair of blocks i and m among $\binom{b}{2}$ possible blocks, use (6) to repeat searching for a pair of treatments t^{ij} in block i and t^{mn} in block m (t^{ij} is not in block m and t^{mn} is not in block i) such that the swap of these two treatments results in the biggest reduction in f_3 without altering the RGD status of the design. If the search is successful, update f_3, NN', and D. If f_3 cannot be reduced further, go to the next pair of blocks. This process is repeated until f_3 reaches its lower bound or there is no further gain by any further possible treatment swap.

The advantage of using f_2 and f_3 instead of E as my objective functions is now clear. First, I do not require a connected starting design in step 1. Second, I do not require the update of NN' to evaluate the derived delta functions in steps 2 and 3. The addition and subtraction operations (in integer mode) used in the evaluations of these derived delta functions are certainly computationally cheaper than those in the updating formulas for C^- used by Jones and Eccleston (1980) and Venables and Eccleston (1993).

I now illustrate the steps of my algorithm for constructing an IBD of size (9, 3, 3):

3 1 7	3 1 9	3 1 9	3 1 9	3 1 9	3 1 9
9 8 4	7 8 4	7 8 4	7 8 4	7 8 4	7 8 4
5 2 6	5 2 6	5 2 6	5 2 6	5 2 6	5 2 6
3 2 7	3 2 7	9 2 7	3 2 7	1 2 7	9 2 7
5 9 8	5 9 8	5 3 8	5 3 8	5 3 8	5 3 8
6 1 4	6 1 4	6 1 4	6 1 4	6 1 4	6 1 4
5 9 4	5 9 4	5 9 4	5 9 4	5 9 4	5 9 4
2 8 3	2 8 3	2 8 3	2 8 9	2 8 9	2 8 1
7 1 6	7 1 6	7 1 6	7 1 6	7 3 6	7 3 6
(A)	(B)	(C)	(D)	(E)	(F)

Step 1 consists of generating the starting design (A). Now $f_2 = 41$. Step 2 consists of (B), (C), (D), and (E), after which f_2 is reduced to 27 and (E) becomes an RGD. Design (B), for example, is obtained by swapping treatment 7 and 9 in block 1 and 2 of (A). Now $f_3 = 29$. Step 3 consists of (F), after which f_3 is reduced to 27. Design (F) is the final design with $E = .7273$. This example shows that RGD's of the same size might not have the same E. Although both (E) and (F) are RGD's of the same size, (F) is optimal and (E) is not.

Because good designs need a blend of theory and computing power (Patterson and Williams 1990), my algorithm exploits the relationship between a two-replicate resolvable design and its *contraction* and a design and its *dual*. A two-replicate resolvable design D of size $(v, k, r) = (ks, k, 2)$ is uniquely determined by its contraction, a nonresolvable symmetrical design D^* of size $(v^*, k^*, r^*) = (s, k, k)$ (John 1987, sec. 4.9; Patterson and Williams 1976b). To construct D from D^* with entries d^*_{ij}, I first generate replicate 1 of D in any order with entries d_{ij}. Now, replicate 2 of D is the rearrangement of the entries in replication 1 such that d_{ij} is in row d^*_{ij}. For example, a two-replicate resolvable design constructed from a nonresolvable symmetrical design of size (9, 3, 3) in my previous illustration is

9	2	17	2	4	8
25	24	13	5	7	3
11	5	6	9	23	1
10	7	20	13	12	15
14	23	22	11	14	16
21	4	12	6	21	18
16	27	15	25	20	19
3	26	8	24	22	26
19	1	18	17	10	27

Because the relation between E of a design D and E^* of its contraction D^* is $E = (v - 1)/\{v - 2s + 1 + 4(s - 1)/E^*\}$, a design is optimal if its contraction is optimal. This result is very handy because to construct a two-replicate resolvable design of size (27, 3, 2) as in my example I only have to construct a nonresolvable symmetrical design of size (9,3,3).

A dual of a design D of size (v, k, r) is a design D' of size $(v', k', r') = (b, r, k)$ obtained by swapping the treatments and blocks symbols in the original designs. For example, the dual of the design (2 7), (2 8), (6 10), (9 4), (3 5), (6 9), (5 7), (1 4), (1 10), (3 8), (3 9), (5 1), (7 6), (10 8), (4 2) is the following design with blocks as columns:

8	1	5	4	5	3	1	2	4	3
9	2	10	8	7	6	7	10	6	9
12	15	11	15	12	13	13	14	11	14.

Because the relation between E of a design D and E' of its dual D' is $E = (v - 1)/\{v - b + (b - 1)/E'\}$ (John 1987, sec. 2.8; Patterson and Williams 1976b), a design is optimal if its dual is optimal. Thus, for designs with $v > b$, my algorithm works with their duals whose concurrence matrix will be of size $b \times b$ instead of $v \times v$. For resolvable designs with $r > 2$ and $v > b$, I also work with their duals. To retain the resolvability of the original design, however, the treatment swaps are restricted to treatments in different blocks but in the same position within a block.

For resolvable designs with $v \leq b$, I repeat my basic algorithm r times. This *sequential* approach was reported by Nguyen (1993).

If the purpose (as in the second example in the Introduction) is to augment an existing design D_0 of

size (v, k, r_0) by a design D of size (v, k, r) to produce an efficient combined design of size $(v, k, r_0 + r)$, my algorithm first forms NN' of D_0 before going to step 1. Here, I borrow the idea from the DETMAX algorithm of Mitchell (1974) according to which blocks in D_0 are *protected*. I use this method to construct a resolvable BIBD of size (18, 6, 17) with ease by adding nine replicates to a resolvable design of size (18, 6, 8), This BIBD was first reported by John (1973). I also use this method sequentially to construct α-resolvable designs.

Several tries are made for each (v, k, r) combination. For non-RGD's, among constructed designs with the smallest f_2, the design with the highest E will be chosen. For RGD's, among constructed designs with the smallest f_3, the design with the highest E will be chosen. For most (v, k, r) combinations in this study, 10 tries are sufficient to find an efficient design.

The *stopping* rule of my algorithm for each (v, k, r) combination is that either the specified number of tries is exhausted or the E of the best constructed design is sufficiently close to the upper bound. The upper bound is $U = \min(U_{WP}, U_J, U_T)$ with U_{WP} being 1 for nonresolvable designs. U_{WP} is the bound of Williams and Patterson (1977) good for any resolvable design with $v \geq b$, U_J is the bound of Jarrett (1989) good for any RGD, and U_T is the bound of Tjur (1990) good for any *binary* design (a design for which no treatment occurs more than once in a block).

Although my described algorithm is meant for IBD's with equal replication, it can be easily modified to handle unequal replication situations. The method of permuting the treatments within blocks of my constructed IBD to produce optimal or near-optimal row-column designs was discussed by Nguyen and Williams (1993a,b).

3. EVALUATION OF THE ALGORITHM

Forty-five of the 63 known BIBD's listed by Fisher and Yates (1963) with $v \leq 100$ and $r, k \leq 10$ were reproduced. Of these 45, 26 were obtained in the first try and 7 required more than 10 tries. The remaining 18 non-BIBD's that my algorithm produces are within .5% of optimality.

Of the 209 (v, k, r) combinations of Mitchell and John (1976) with $3 \leq v \leq 12$ and $2 \leq k, r \leq 10$, 200 RGD's were obtained. All except 14 constructed designs match the tabulated E. Of these 14, 7 have E .001 or more, higher than the tabulated E, and 7 have E .001 or more, lower than the tabulated E. These 7 require more than 10 tries to match the tabulated E. There are 7 combinations that Mitchell and John (1976) failed to handle—namely, (v, k, r) = (12, 2, 5), (12, 2, 6), (12, 3, 3), (12, 3, 8), (12, 4, 9), (12, 6, 10), and (12, 9, 9). The E's of our RGD's for these combinations are .504, .524, .678, .721, .816, .908, and .969, respectively. The time required for these 216 combinations each with 10 tries on my 33-megahertz 486DX PC is $5\frac{1}{2}$ minutes. One hundred and seventy-eight combinations have the best tries as the first tries.

Of the 188 (v, k, r) combinations of Hall and Jarrett (1981) with $10 \leq v \leq 60$, $2 \leq r < k \leq 10$, and $r \leq 5$, I could construct 93 RGD's, but these authors produced 95 RGD's. The three combinations for which I failed to find RGD solutions are (50, 8, 4), (52, 8, 4), and (54, 8, 4). I found an RGD solution for the combination (60, 9, 3). One hundred and seventy-three constructed designs are within 1% of optimality (77 are optimal). Of the remaining 17, 2 required more than 10 tries to be within 3% of optimality. Like GCD's, my designs show an improvement in E over the fractional cylic designs of John, Wolock, and David (1972). The time required for these 188 combinations is $2\frac{3}{4}$ hours on my PC.

The 414 (v, k, r) resolvable combinations of Patterson and Williams (1976a) satisfy the conditions $v \leq 100$, $r = 2, 3, 4$, $r \leq k \leq 16$, and $v = ks$ $(s > 1)$. Of the 198 combinations with $k \leq s$, I could construct 193 RGD's, but 188 RGD's were obtained by the α algorithm. One hundred and ninety-seven constructed designs are within 1% of optimality (44 are optimal). The worst design has E reaching 98.92% of the upper bound. Of the 159 combinations with $s < k \leq s^2$, 102 are optimal. Of the remaining, the worst has E reaching 99.86% of the upper bound. All constructed designs for the 57 combinations with $k > s^2$ are optimal. Overall, these results slightly improve those of Nguyen (1993). The time required for the 198 combinations with $k \leq s$ is about 15 hours, for the 159 combinations with $s < k \leq s^2$ is 39 minutes and for the 57 combinations with $k > s^2$ is $\frac{1}{2}$ minute.

The differences in E between our designs and GCD's and α designs are marginal for the combinations within the range under study. This is comprehensible because, within this range, these GCD's and α designs are either optimal or near-optimal. Most GCD's are within 3% of optimality and most α designs are within 1% of optimality. As such, my algorithm is more useful in situations in which there is no cyclic solution or there are more efficient noncyclic solutions. For example, the E of the cyclic solutions of the three combinations (12, 3, 6), (14, 3, 6), and (14, 5, 10) in table 3 of Hall and Jarrett (1981) are .7133, .7044, and .8601. These solutions are not RGD's. The E of my corresponding optimal RGD solutions are .7230, .7137, and .8611. For the resolvable combinations (30, 5, 4) and (36, 6, 4) in which lattice solutions are not available, the E of the α designs in table 1 of

Patterson and Williams (1976a) are .8046 and .8360. The E of my resolvable solutions are .8053 and .8393.

The running time of my algorithm for individual IBD depends on the size of (v, k, r) and whether the design should be resolvable, but overall it compares favorably with other general block design algorithms in the literature. For example, for an IBD of size (60, 9, 3), an optimal solution with $E = .8786$ is obtained in 12 out of 100 tries of my algorithm. The average time per try for this combination is 47 seconds on my PC. For a resolvable IBD of size (98,7,2), an optimal solution with $E = .7614$ is obtained in five out of 100 tries of our algorithm. The average time per try for this combination is .9 seconds.

4. APPLICATIONS

In this section, I shall illustrate how my algorithm can be applied to the experimental situations discussed in the Introduction. In the first problem, 14 chemical treatments are tested for efficacy on 28 AC panels with each panel divided into 5 strips and each treatment applied to 10 strips. The following design of size (14, 5, 10) wih blocks as rows is my solution to this problem. This design is optimal with $E = .8611$. It can be verified that the number of times a pair of treatments occurs in the same panel is either 3 or 4, no matter which pair of treatments is selected. Hence the design is also an RGD:

10	5	13	2	7
1	2	7	14	12
2	3	6	9	8
8	7	12	4	14
13	1	8	2	9
14	4	9	5	6
12	8	13	6	10
10	4	2	3	12
4	5	11	1	8
11	6	12	5	7
3	1	11	14	13
7	3	11	13	4
6	3	7	1	8
14	5	9	13	10
14	9	7	11	2
11	7	13	6	9
2	4	5	8	11
3	5	12	9	1
3	9	12	13	4
1	6	14	4	10
2	12	10	11	6
4	9	7	8	10
14	5	3	6	2
8	5	12	13	14
11	10	12	1	9
1	2	4	6	13
1	5	10	3	7
11	14	3	8	10

The second problem requires a replicate of the 15 treatments to be added to a design of size (15, 3, 2). This design corresponds to an experiment in which 15 brands of paint were tested with a UV weatherometer that can only handle three paint panels at a time, and there were only two replicates for each paint. The last five blocks (columns) of the following design render the solution for this problem. The new design of size (15, 3, 3) is an optimal RGD with $E = .6604$. It can be verified that the number of times a pair of treatments occurs in the same panel is either 0 or 1:

12	2	5	15	12	13	13	10	11	3	8	11	12	5	14
9	1	11	8	5	3	7	14	6	14	13	1	2	15	7
8	15	10	4	7	6	1	2	4	9	10	9	6	3	4.

In the third problem, 21 timber species are compared with respect to service life in 35 blocks of size 6 with each species replicated 10 times. The blocks are divided into five subsets with each subset having two replications of the treatments. The following two-resolvable design of size (21, 6, 10) with blocks as rows is my solution to this problem. Again, my solution is an RGD with $E = .8733$, which is 99.98% of the upper bound. This compares favorably with a corresponding two-resolvable PBIBD in plan **T69** of Clatworthy (1973) with $E = .82$. The number of times a pair of timber species occurs in the same block is either 0 or 5 in the PBIBD and either 2 or 3 in my solution:

7	21	18	14	3	10
17	7	12	2	19	11
11	15	6	9	20	14
16	6	4	18	8	12
17	13	4	1	10	20
15	5	8	2	21	13
9	5	3	19	1	16
1	11	5	14	13	12
9	18	7	20	8	5
11	3	2	9	21	4
19	13	7	4	15	16
17	2	14	16	10	8
17	15	3	6	1	18
10	6	20	12	21	19
9	15	8	10	1	12
21	20	7	16	1	11
15	2	14	18	19	20
4	7	10	5	2	6

16	21	17	18	9	13
3	6	8	11	13	19
17	5	12	3	4	14
14	1	21	4	19	8
10	15	17	5	16	11
13	3	16	20	12	2
9	17	13	7	14	6
2	5	21	6	18	1
18	10	11	4	19	9
3	12	15	8	20	7
2	1	13	12	19	18
8	7	1	10	11	14
3	15	18	11	16	4
21	6	16	14	12	9
20	5	6	3	10	13
5	21	19	17	7	15
2	17	20	9	8	4

5. CONCLUSION

In this article, I have described an algorithm for constructing block designs and have given evidence that it works well in various experimental situations. I trust that my algorithm will be a useful and user-friendly addition to the toolkit of designers of experiments, particularly those with a new philosophy of designing experiments; that is, "Design for the experiment, do not experiment for the design" (Federer 1990; Cook and Nachtsheim 1990). The algorithm is implemented in a PASCAL program named BIB and is available from me.

ACKNOWLEDGMENTS

I thank R. G. Jarrett, J.A. John, E. R. Williams, the referees, and the associate editor for valuable comments and suggestions. This article is dedicated to the memory of Toby J. Mitchell whose DETMAX algorithm has inspired much of my work.

[*Received January 1993. Revised February 1994.*]

REFERENCES

Clatworthy, W. H. (1973), *Tables of Two-Associate-Class Partially Balanced Designs* (Applied Mathematics Ser. 63), Washington: National Bureau of Standards.

Cook, R. D., and Nachtsheim, C. J. (1990), "Response to James M. Lucas" (in "Letters to the Editor"), *Technometrics*, 32, 364–365.

Eccleston, J. A., and Hedayat, A. (1974), "On the Theory of Connected Designs: Characterization and Optimality," *The Annals of Statistics*, 2, 1238–1255.

Federer, W. T. (1990), Discussion of "The Non-Orthogonal Design of Experiments," by R. Mead, *Journal of the Royal Statistical Society*, Ser. A, 153, 187.

Fisher, R. A., and Yates, F. (1963), *Statistical Tables for Biological Agricultural and Medical Research*, London: Oliver & Boyd.

Hall, W. B., and Jarrett, R. G. (1981), "Nonresolvable Incomplete Block Designs With Few Replications," *Biometrika*, 68, 617–627.

Jarrett, R. G. (1989), "A Review of Bounds for the Efficiency Factor of Block Designs," *Australian Journal of Statistics*, 31, 118–129.

John, J. A. (1987), *Cyclic Designs*, New York: Chapman & Hall.

John, J. A., and Mitchell, T. J. (1977), "Optimal Incomplete Block Designs," *Journal of the Royal Statistical Society*, Ser. B, 39, 39–43.

John, J. A., Wolock, F. W., and David, H. A. (1972), *Cyclic Designs* (Applied Mathematics Ser. 62), Washington: National Bureau of Standards.

John, P. W. M. (1973), "A Balanced Design for Eighteen Varieties," *Technometrics*, 15, 641–642.

Jones, B. (1976), "An Algorithm for Deriving Optimal Block Designs," *Technometrics*, 18, 451–458.

Jones, B., and Eccleston, J. A. (1980), "Exchange and Interchange Procedures to Search for Optimal Designs," *Journal of the Royal Statistical Society*, Ser. B, 42, 238–243.

Kiefer, J. (1959), "Optimum Experimental Design," *Journal of the Royal Statistical Society*, Ser. B, 21, 272–319.

Lamacraft, R. R., and Hall, W. B. (1982), "Tables of Cyclic Incomplete Block Designs With $r = k$," *Australian Journal of Statistics*, 24, 350–360.

Lin, S., and Kernighan, B. W. (1973), "An Effective Heuristic Algorithm for the Travelling-Salesman Problem," *Operations Research*, 21, 498–516.

Mitchell, T. J. (1973), "Computer Construction of Small '*D*-Optimal' Incomplete Block Designs," *Proceedings of the 39th Session of the ISI*, 2, 199–205.

——— (1974), "An Algorithm for Construction of '*D*-Optimal' Experimental Designs," *Technometrics*, 16, 203–210.

Mitchell, T. J., and John, J. A. (1976), *Optimal Incomplete Block Designs* (ORNL/CSD-8), Washington: National Technical Information Service, U.S. Dept. of Commerce.

Nguyen, N.-K. (1993), "An Algorithm for Constructing Optimal Resolvable Incomplete Block Designs," *Communications in Statistics, Part B—Simulation and Computation*, 22, 911–923.

Nguyen, N.-K., and Dey, A. (1990), "Computer-Aided Construction of Small (M, S)-Optimal Incomplete Block Designs," *Australian Journal of Statistics*, 32, 399–410.

Nguyen, N.-K., and Miller, A. J. (1992), "A Review of Some Exchange Algorithms for Constructing Discrete *D*-Optimal Designs," *Computational Statistics & Data Analysis*, 14, 489–498.

Nguyen, N.-K., and Williams, E. R. (1993a), "An Algorithm for Constructing Optimal Resolvable Row-Column Designs," *Australian Journal of Statistics*, 35, 363–370.

——— (1993b), "Construction of Row-Column Designs by Computer," Technical Report 93/53T, CSIRO-IAPP Biometrics Unit, Clayton, Victoria, Australia.

Paterson, L. J. (1983), "Circuits and Efficiency Factors in Incomplete Block Designs," *Biometrika*, 70, 215–226.

Paterson, L. J., and Patterson, H. D. (1983), "An Algorithm for Constructing α-Lattice Designs," *Ars Combinatoria*, 16A, 87–98.

Patterson, H. D., and Williams, E. R. (1976a), "A New Class of Resolvable Incomplete Block Designs," *Biometrika*, 63, 83–92.

——— (1976b), "Some Theoretical Results on General Block Designs," *Proceedings of the 5th British Combinatorial Conference*, (Congressus Numeratium XV), Winnipeg: Utilitas Mathematica, pp. 489–496.

——— (1990), Discussion of "The Non-Orthogonal Design of Experiments," by R. Mead, *Journal of the Royal Statistical Society*, Ser. A, 153, 185–186.

Russell, K. G., Eccleston, J. A., and Knudsen, G. J. (1981), "Algorithms for Construction of (M, S)-Optimal Block Designs

and Row-Column Designs," *Journal of Statistical Computation and Simulation*, 12, 93–105.

Shah, K. R. (1960), "Optimal Criteria for Incomplete Block Designs," *The Annals of Mathematical Statistics*, 31, 791–794.

Tjur, T. (1990), "A New Upper Bound for the Efficiency Factor of a Block Design," *Australian Journal of Statistics*, 32, 231–237.

Venables, W. N., and Eccleston, J. A. (1993), "Randomized Search Strategies for Finding Near-Optimal Block and Row-Column Designs," *Australian Journal of Statistics*, 35, 371–382.

Whitaker, D., Triggs, C. M., and John, J. A. (1990), "Construction of Block Designs Using Mathematical Programming," *Journal of the Royal Statistical Society*, Ser. B, 52, 497–503.

Williams, E. R., and Patterson, H. D. (1977), "Upper Bound for Efficiency Factors in Block Designs," *Australian Journal of Statistics*, 19, 194–201.