Statistical Methods & Applications
https://doi.org/10.1007/s10260-026-00835-2

ORIGINAL PAPER

Heuristic optimization of Balanced Incomplete Block Designs under practical constraints

Alberto Lombardo[1] · Giulia Marcon[1]

Received: 21 July 2025 / Accepted: 18 January 2026

Abstract

An approach to the construction of Balanced Incomplete Block Designs (BIBD) is described. The exact pairwise balance of treatments within blocks (second-order balancing condition) is required by standard BIBD. This requirement is attainable when λ is an integer, namely when the relationship among the number of treatments t, blocks b, and block size k allows for an equal number of co-occurrences for every treatment pair. This work presents an algorithm for generating quasi-BIBD with particular attention to settings where a second blocking variable is taken into account (Youden squares) and the blocks are assigned to s groups (sessions) where all treatments are equally represented inside each one. Unlike classical resolvable or nested BIBDs, which require an exact combinatorial solution and therefore an integer value of λ, the designs developed here explicitly allow λ to be non-integer and do not rely on the existence of an exact BIBD. Two real-world applications are presented. The first refers to an exploratory experiment with 8 two-level factors leading to t=20 trials, assessed by k=4 evaluators through b=20 blocks divided into s=4 balanced sessions. The second example refers to a robust design experiment based on a Central Composite Design involving 4 technological and 2 environmental factors leading to t=30 trials, evaluated in k=5 environmental conditions within b=30 blocks divided into s=5 balanced sessions. In these experimental settings, the second-order balancing condition is not attainable since λ is not an integer. The proposed algorithm has been designed to approximate this condition as closely as possible, maintaining the first-order balancing condition, highest D-optimality and connectedness.

Keywords BIBD · Design Of Experiments · Heuristic Optimization · Balancing · Block design · Connectedness

Extended author information available on the last page of the article

Published online: 25 February 2026

1 Introduction

A central objective in the design of experiments is the construction of efficient designs that significantly reduce the number of experimental runs compared to full factorial structures, while still preserving the ability to extract meaningful information about the system under study. The advantages of such reductions are particularly in terms of time, cost, and resource savings (Box et al. 2005). However, these benefits must be carefully weighed against the potential loss of estimability or interpretability.

Ensuring that the effects of primary interest, typically main effects and sometimes second order factors (their pairwise interactions), remain estimable is a fundamental requirement. This is generally verified by transforming factor levels into pseudo-regressors, whose number corresponds to the degrees of freedom associated with each effect (i.e., the number of levels minus one for main effects, and the product of these for interactions). Several coding schemes are available: dummy coding assigns, for each factor level, an indicator variable taking value 1 when the treatment is present and 0 otherwise; alternatively, contrast coding such as $(-1, 0, -1)$ enforce the zero-sum property across columns and are widely used in the construction of balanced and orthogonal designs (Wynn 1985). Another option is to employ orthogonal polynomials, which provide mutually uncorrelated regressors for ordered factor levels (Draper and Smith 1998). The latter are particularly advantageous in full factorial designs, where orthogonality holds not only between pseudo-regressors related to different factors, but also among contrast components within each factor. The full-rank condition of the resulting design matrix confirms the estimability of the modelled effects and also forms the basis for evaluating the design optimality when comparing full rank competing designs. In our context, the contrast coding parametrisation is adopted to construct the pseudo-regression matrix used in the evaluation of the determinant D.

Among the most widely used criteria for design comparisons, the D-optimality favours designs whose determinant of the information matrix is largest, thereby minimizing the generalized variance of the regression coefficients estimated (Wu and Hamada 2011). Alternative optimality measures, which focus on the maximization of the smallest eigenvalue or the minimization of the largest, are also used in specific contexts. Full factorial designs are D-optimal by construction, as the associated design matrix (of the principal factors and of all interactions up to the highest order) is always diagonal. Worthy of particular and practical interest are orthogonal reduced designs, which retain the orthogonality property for effects of interest (order I or even II, i.e. main factors or pairwise interactions) despite the reduction. For this reason, significant theoretical developments have focused on constructing such reduced yet orthogonal designs wherever possible.

The theory for symmetric designs (in which all factors have the same number of levels) is well developed. In particular, in the case of two-level factors, where each effect is associated with a single degree of freedom, and more generally, with the number of levels k being a prime or a power of a prime, the correspondence between the design matrix and estimability is straightforward, e.g., algebraic techniques based on Galois fields, Giovagnoli (1977). The methodology is relatively simple when k is prime and slightly more complex when it is a prime power. In contrast, the construction of asymmetric designs is more challenging. In such cases, the properties above and algebraic techniques that facilitate reductions of the complete symmetric design can not be longer performed, and alternative methods must be employed.

The paper is organized as follows. Section 2 provides a brief review of Balanced Incomplete Block Designs and the challenges involved in achieving first- and second-order balance conditions. The introduction of an additional block, named *session*, is presented in section 2.1, while the connectedness properties of the proposed designs are discussed in section 2.2. Section 3 discusses the limitations of traditional optimization criteria and introduces heuristic approaches tailored to experimental constraints. A proposal algorithm is also described. Section 4 presents two illustrative examples from real-world contexts, demonstrating the need for flexible design strategies. Section 5 discusses practical considerations and limitations.

2 Balanced Incomplete Block Designs

An important and extensively studied class of experimental designs is that of Balanced Incomplete Block Designs (BIBD). These designs are especially useful when it is not feasible to include all treatment levels within a single block. In the standard BIBD framework, a primary treatment factor is considered at t levels, while each block can accommodate only $k < t$ treatments. The goal is to construct a set of b blocks such that each treatment appears exactly r times across all blocks (we named *first-order balance*). This condition implies that the total number of experimental runs satisfies the identity $n = tr = bk$. In addition to equal replication, it is necessary to constrain the set-up in order to avoid that in a block a greater number with the same treatment occurs. This could invalidate the sampling, as the confounding between treatments and blocks. At this aim is desirable that each pair of treatments co-occur in the same number of blocks. This number, denoted λ, ensures a property we named *second-order balance*. This imposes the constraint

$$\lambda = \frac{b\begin{pmatrix} k \\ 2 \end{pmatrix}}{\begin{pmatrix} t \\ 2 \end{pmatrix}} = \frac{b\,k\,(k-1)}{t\,(t-1)} \tag{1}$$

reflecting the requirement to allocate all $\begin{pmatrix} t \\ 2 \end{pmatrix}$ unique treatment pairs across the b blocks, each of which contains $\begin{pmatrix} k \\ 2 \end{pmatrix}$ such pairs. In principle, exact second-order balance can be achieved when $b = \begin{pmatrix} t \\ k \end{pmatrix}$, but in practice, this quickly becomes infeasible as k approaches $t/2$, due to the combinatorial explosion in the number of required blocks, and consequently in the number of runs. However, certain configurations allow the construction of BIBD with a reduced number of blocks while still satisfying the second-order balance conditions.

Designs of particular interest include those in which a third factor is incorporated, representing the position of a treatment within the block. Such designs can be created when the number of treatments t equals, or is a multiple of, the number of blocks b, thus it becomes possible to construct structures known as Youden squares. Despite their name, these are not square matrices but can be considered as incomplete Latin squares.

An example concerns the design where $t = b = 5, r = k = 4$ and consequently $n = 20$. As can be seen in Table 1, treatment pairs such as AB and AC appear together in exactly three blocks, as expected under the second-order balance condition (AB occurs in each block only three times, in 2, 4 and 5; AC occurs three times, in 1, 2 and 5), and so on for all the pairs. In this case, the corresponding value of λ is given by

$$\lambda = \frac{5\begin{pmatrix} 4 \\ 2 \end{pmatrix}}{\begin{pmatrix} 5 \\ 2 \end{pmatrix}} = \frac{\frac{5\cdot4\cdot3}{2}}{\frac{5\cdot4}{2}} = \frac{30}{10} = 3.$$

Such designs cannot be obtained through algebraic constructions, therefore, in these cases, heuristic algorithms and computational search techniques are essential for identifying feasible configurations.

Recent literature has explored various approaches to constructing Balanced Incomplete Block Designs (BIBDs), particularly when exact combinatorial conditions are hard to satisfy or must be extended to more complex experimental contexts. Among these, Rueda et al. (2020) propose a memetic algorithm for generating approximate BIBDs, Agashe et al. (2024) introduce a new class of BIBDs that extend classical BIBDs into multilevel block structures, Benedict et al. (2023) investigate the construction of non-symmetric BIBDs derived from symmetric configurations, Chisaki et al. (2025) propose Spanning Bipartite Block Designs. While similar in motivation, our proposal addresses the specific challenge of simultaneously satisfying first-order balance, approximate second-order balance, and session-level replication using a heuristic strategy.

In practical applications, the exact second-order balance required by a standard BIBD is often infeasible because the concurrence parameter λ is not an integer. In such cases, experimental designs can still preserve the first-order balance while approximating the second-order one as closely as possible. To formalize these cases, we adopt the following terminology. An *optimal BIBD* denotes a design that exactly satisfies both first- and second-order balance conditions, corresponding to integer λ.

Table 1 Youden Square. ($t = b = 5,\ k = r = 4,\ \lambda = 3$) Design (left). Treatment pairs in the same block (centre). Number of occurrences of pairs in the same block (right)

Blocks	I	II	III	IV
1	D	E	C	A
2	E	A	B	C
3	C	B	D	E
4	A	D	E	B
5	B	C	A	D

DE	CD	AD	CE	AE	AC
AE	BE	CE	AB	AC	BC
BC	CD	CE	BD	BE	DE
AD	AE	AB	DE	BD	BE
BC	AB	BD	AC	CD	AD

	B	C	D	E
A	3	3	3	3
B		3	3	3
C			3	3
D				3

When λ is non-integer and exact pairwise balancing cannot be achieved, a design that maintains first-order balance but only approximates the second-order condition is referred to as a *quasi-BIBD*. Among these, configurations that minimize deviations from perfect pairwise balance while maximizing inferential efficiency (e.g., *D*-optimality) are termed *near-optimal BIBD*. These last two categories represent the focus of the heuristic strategy proposed in this paper.

2.1 Nested block designs with *session* control

A particularly relevant feature in practical applications is the possibility of dividing the execution of blocks into separate groups or *sessions*. This becomes essential when the total number of blocks is large and cannot be executed simultaneously (e.g. when blocks must be distributed across different days or shifts due to logistical constraints). This introduces an additional layer of structure to the design that must be explicitly accounted for to ensure statistical validity. In such cases, it is desirable to ensure that the session factor is also orthogonal with respect to the treatments: when the number of blocks b is a multiple of the number of treatments t, it is possible to organize the blocks into s sessions such that each treatment appears equally often in each session. This structure facilitates session-wise treatment balance, which is crucial in avoiding confounding effects between treatment mean responses and session-level variation. This guarantees that any systematic effect introduced by the session (e.g., fatigue, changes in environmental conditions, or subjective bias) is distributed equally across all treatments, thus preserving the integrity of the comparisons. In the analysis of variance (ANOVA), this session structure necessitates explicit nesting of blocks within sessions, protecting treatment comparisons from such confounding, and may also support the modeling of session-specific trends if temporal effects are suspected.

To formalize this, consider the case where $t = b$, and each treatment is replicated $r = k$ times. If the total number of blocks b is divisible by the block size k, then the number of blocks per session is $m = t/k$, and the total number of sessions becomes $s = t/m = k$. Each session thus contains m blocks and provides a complete and balanced representation of all treatments. This representation can also be viewed as a generalization of Youden squares, in which rows correspond to sessions and columns to evaluators, ensuring equal treatment representation across both dimensions.

The concept of introducing a session factor extends classical notions of resolvable and nested block designs. In the terminology of Calinski and Kageyama (2012), sessions would be classified as *superblocks*. However, within the framework proposed here, sessions are not exact replicates of the block structure. Their composition differs from one session to another, because the procedure aims to achieve first-order balance when considering the full design as a whole. Position-wise balance is therefore not enforced within each individual session, but is attained across the complete set of blocks. In *resolvable BIBDs*, blocks can be partitioned into groups such that each treatment appears exactly once per group (Abel et al. 2006). The session factor proposed in this work plays a role that is reminiscent of such replicates, since treatments are required to appear equally often within each session, but the analogy remains conceptual rather than combinatorial. Sessions do not constitute full copies of the design, nor are they constrained to reproduce the classical structure of a replicate; instead, they arise from practical considerations such as the temporal organisation of blocks or evaluator availability. The goal is not to enforce a formal resolution of the block set, but to guarantee that session-level variation is orthogonal to treatment effects. Similarly, in *nested BIBDs* blocks at one level are formally contained within larger blocks according to a fixed hierarchy, and the combinatorial relations among blocks are determined by this nesting structure (Buratti et al. 2024; Morgan et al. 2001). Although the resulting layout may resemble a two-level structure, sessions do not function as super-blocks in the combinatorial sense, and the design does not impose the constraints typical of nested constructions. A further and more fundamental difference concerns the value of λ. Classical resolvable or nested BIBDs presuppose the existence of an exact BIBD and thus require λ to be an integer. The designs proposed in this work explicitly accommodate cases in which λ is non-integer and no exact BIBD exists. In these settings, the session factor coexists with a heuristic search that constructs quasi-BIBDs satisfying first-order balance and approximating second-order balance as closely as possible. This combination of session-wise balance and approximate pairwise balance lies outside the scope of both resolvable and nested frameworks. From this perspective, the proposed structure can be viewed as a practical generalisation of resolvable designs: whenever exact replicates could exist, the session framework is compatible with them, but when combinatorial conditions fail, it still ensures balanced treatment representation across sessions. At the same time, it may also be interpreted as a packing-type configuration enriched with a session constraint and optimised for statistical efficiency, rather than as a purely combinatorial object.

Specifically, the experimental constraints considered in this work are the following:

- The block size k, which specifies how many positions must be filled and corresponds, in our examples, to the number of evaluators; within each block, each treatment can be assigned at most once across the k evaluators;
- The subset of treatments that, although theoretically repeatable within a block, are restricted from occurring more than once;
- The maximum number of blocks that can be conducted within a single session, which requires grouping blocks so that each treatment has the same number of replicates within each session. By contrast, the total number of blocks b is not considered limiting, and therefore it is set equal to the number of treatments. Designs with $b < t$ or $b > t$ are not considered in this work.

To illustrate how these design principles and session structures can be effectively applied in practical scenarios, two representative examples are presented in section 4.

2.2 Connectedness

Connectedness is a fundamental property for block designs because it guarantees that all treatment contrasts are estimable: treatments must be linked through chains of co-occurrences within blocks (or sessions), ensuring that all treatment contrasts remain estimable in the associated linear model. Connectedness can be checked combinatorially by constructing the treatment-association graph (vertices = treatments; an edge between two treatments if they appear together in at least one block) and verifying that this graph is connected. If the graph induced by the BIBD is weakly connected, the design is globally connected in the classical statistical sense, and full estimability of treatment effects is guaranteed. Formally, let $G = (V, E)$ be the undirected treatment–association graph of a block design, where $V = \{1, \ldots, v\}$ is the set of treatments and $(i, j) \in E$ if and only if treatments i and j co-occur in at least one block. The design is globally connected if and only if G consists of a single connected component. This requires that $\forall i, j \in V, \exists (i = v_0, v_1, \ldots, v_k = j)$ such that $(v_\ell, v_{\ell+1}) \in E, \forall \ell = 0, \ldots, k-1$. Equivalently, letting A denote the $v \times b$ incidence matrix of the design and $\mathrm{C} = \mathrm{AA}^\top$ its concurrence matrix, global connectedness holds if and only if rank(C) = $v - 1$ or, equivalently, C has a single zero eigenvalue associated with the invariance to adding a constant to all treatment effects. Under this condition, all treatment contrasts are estimable in the associated linear model.

Definition 1 **Connectedness** (Eccleston and Hedayat 1974). The treatments T_i and T_j are connected if there exists a chain $T_i\, B_1\, T_{i_1}\, B_2\, T_{i_2} \ldots T_{i_{h-1}}\, B_h T_{i_h}\, B_{h+1} T_j$. In this case, an unbiased estimator of $T_i - T_j$ is obtained from this chain. Two treatments T_i and T_j, $i \neq j$, of a block design are said to be *globally connected* if each replicate of T_i is connected by a chain, as defined by Bose (1947), to each replicate of T_j. Two treatments T_i and T_j, $i \neq j$, of a block design are said to be *pseudo-globally connected* if each replicate of T_i is connected by a chain, as defined by Bose (1947), to at least one replicate of T_j and vice versa.

A straightforward strategy for constructing a design with $t = b = 20$ and $k = r = 4$ is to stack four copies of the well-known $t = b = 5$ and $k = r = 4$ design, which is easy to generate and corresponds to a classical Youden square structure when block positions are retained as an additional blocking factor (see Table 2). Although this construction preserves the positional structure, it lacks global connectedness: when the treatment–association graph is computed, the graph decomposes into four disconnected components (see Fig. 1). From an experimental perspective, this implies that block and treatment effects cannot be estimated simultaneously, because the incidence matrix does not achieve full rank.

An alternative approach treats the four constituent sub-designs as distinct sessions, with blocks and treatments nested within sessions in groups of five. Under this parametrization, both block and treatment effects become estimable; however, three degrees of freedom are lost due to the introduction of the session factor. Thus, instead of the full 19 degrees of freedom available for treatments in a globally connected design with $t = 20$, only 16 remain once sessions are incorporated.

In contrast, the design proposed in this work exhibits substantially improved inferential properties (see Tables 5 and 7 in the Appendix):

- It is globally connected: the treatment–association graph consists of a single connected component;
- When sessions are not part of the model, the incidence structure has full rank, so all degrees of freedom are retained: 19 for treatments and blocks, and 3 for block

Table 2 Stacked Youden Sqaures. ($t = b = 20$, $k = r = 4$, $\lambda = 0.63$) Design (table on the top-left). Treatment pairs in the same block (table on the top-right). Number of occurrences of pairs in the same block (table on the bottom)

Blocks	I	II	III	IV	S
1	A	B	C	D	1
2	B	C	D	E	1
3	C	D	E	A	1
4	D	E	A	B	1
5	E	A	B	C	1
6	F	G	H	I	2
7	G	H	I	J	2
8	H	I	J	F	2
9	I	J	F	G	2
10	J	F	G	H	2
11	K	L	M	N	3
12	L	M	N	O	3
13	M	N	O	K	3
14	N	O	K	L	3
15	O	K	L	M	3
16	P	Q	R	S	4
17	Q	R	S	T	4
18	R	S	T	P	4
19	S	T	P	Q	4
20	T	P	Q	R	4

AB	AC	AD	BC	BD	CD
BC	BD	BE	CD	CE	DE
CD	CE	AC	DE	AD	AE
DE	AD	BD	AE	BE	AB
AE	BE	CE	AB	AC	BC
FG	FH	FI	GH	GI	HI
GH	GI	GJ	HI	HJ	IJ
HI	HJ	FH	IJ	FI	FJ
IJ	FI	GI	FJ	GJ	FG
FJ	GJ	HJ	FG	FH	GH
KL	KM	KN	LM	LN	MN
LM	LN	LO	MN	MO	NO
MN	MO	KM	NO	KN	KO
NO	KN	LN	KO	LO	KL
KO	LO	MO	KL	KM	LM
PQ	PR	PS	QR	QS	RS
QR	QS	QT	RS	RT	ST
RS	RT	PR	ST	PS	PT
ST	PS	QS	PT	QT	PQ
PT	QT	RT	PQ	PR	QR

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B		3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C			3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D				3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F						3	3	3	3	0	0	0	0	0	0	0	0	0	0
G							3	3	3	0	0	0	0	0	0	0	0	0	0
H								3	3	0	0	0	0	0	0	0	0	0	0
I									3	0	0	0	0	0	0	0	0	0	0
J										0	0	0	0	0	0	0	0	0	0
K											3	3	3	3	0	0	0	0	0
L												3	3	3	0	0	0	0	0
M													3	3	0	0	0	0	0
N														3	0	0	0	0	0
O															0	0	0	0	0
P																3	3	3	3
Q																	3	3	3
R																		3	3
S																			3

position (Youden–square structure);

- When the design is instead executed across multiple sessions, the session factor absorbs 3 degrees of freedom from the block effects, but, unlike the stacked construction, treatment effects remain fully estimable with 19 degrees of freedom.

Therefore the proposed algorithm provides connected designs where all pairwise contrasts among treatments are estimable (subject to the usual sum-to-zero constraint). The twofold verification (graph-theoretic and linear-algebraic) provides a robust confirmation of estimability and should be sufficient for reproducibility; the detailed results are reported in the Appendix A.

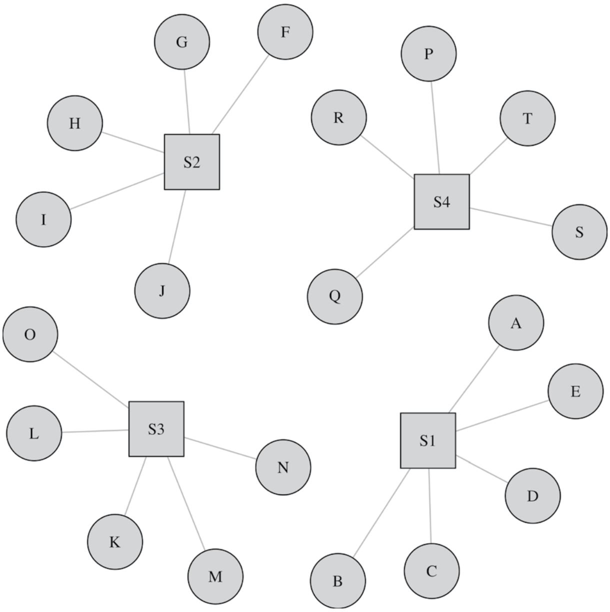

Fig. 1 Not-Connected design in Table 2

3 Heuristic techniques to design optimization

Heuristic methods for experimental design optimization typically rely on selecting a subset of n rows from a full design, consisting of N rows, in such a way that the resulting design satisfies specific statistical or structural criteria. The available procedures start with a subset, chosen randomly or with any other criterion, and iteratively improves it by exchanging one or more rows with those in the $N - n$ remaining design. Each proposed exchange is evaluated using an objective function: if the exchange improves the objective function is accepted; otherwise is discarded. The procedure terminates when either a predefined number of iterations has been reached or no further improvements can be identified. The objective function usually coincides with the determinant of the pseudo-regressor matrix, as previously described, in accordance with D-optimality criteria (Draper and Smith 1998; Wu and Hamada 2011).

While effective in optimizing estimation precision, this class of methods does not typically ensure first or second-order balance conditions. These balance properties are essential in many experimental settings, especially where systematic biases must be minimized, such as in sensory studies, clinical trials, or agricultural experiments. To address this gap, Barone and Lombardo (Barone and Lombardo 2006a, b) proposed a novel heuristic strategy, based on the new criterion, the B-optimality. The algorithm begins with a reduced design that satisfies first-order balance condition, either through algebraic construction or random sampling under constraints. Their approach is based on swapping elements within the same column, and not swapping the entire row. It then optimizes the co-occurrence frequencies of treatment pairs while ensuring, by construction, that first-order balance is preserved at every step. Unlike traditional D-optimality based approaches, this method guarantees that first-order balance is maintained throughout the search process, while progressively approximating second-order balance. Building on the same heuristic framework, our proposal extends their approach to more complex settings by incorporating session structure, offering a practical solution where algebraic methods result inadequate. The proposal algorithm, described in the following section, was implemented in R (Team 2025).

3.1 An algorithm for constructing near-optimal BIBD

Most existing BIBD-focused routines are designed around fixed combinatorial constructions or heuristic optimization that prioritize classical optimality criteria. These approaches are rigid and operate on complete blocks or full-row exchanges, assuming ideal parameters (i.e., when exact BIBDs exist) and often failing or producing suboptimal results when exact configurations are not theoretically possible. Due to the absence of dedicated procedures for generating quasi-BIBD in standard statistical software (e.g., Minitab, R), this heuristic was necessary to manually construct a valid design. Repeated runs of random assignment did not yield satisfactory configurations, demonstrating the need for a principled search mechanism. The algorithm proposed is described in 1 and relaxes the constraint of exact BIBD existence (i.e., allowing for not-integer λ).

Let t be the number of treatments, b the number of blocks, k the number of treatments per block, r the number of replications per treatment, s the number of sessions (if applicable), $\mathcal{T} = \{T_1, \ldots, T_t\}$ the set of treatments, $\mathcal{B} = \{B_1, \ldots, B_b\}$ the set of blocks, $\mathcal{S} = \{S_1, \ldots, S_s\}$ the set of sessions.

We define the assignment matrix:

$$\mathrm{A} \in \{0,1\}^{t \times b}, \quad \text{where } A_{T,B} = \begin{cases} 1 & \text{if treatment } T \text{ is assigned to block } B \\ 0 & \text{otherwise} \end{cases}$$

The heuristic search algorithm for generating an optimal or near-optimal quasi-BIBD is structured to satisfy the following constraints:

(C1) First-Order Balance

Each treatment appears exactly r times:

$$\sum_{B \in \mathcal{B}} A_{T,B} = r \quad \forall T \in \mathcal{T}$$

(C2) Approximate Second-Order Balance

The co-occurrence matrix is defined as follows:

$$\mathrm{C} = \mathrm{A}\mathrm{A}^\top \in \mathbb{N}^{t \times t}, \quad C_{i,j} = \sum_{B \in \mathcal{B}} A_{T_i,B} A_{T_j,B}, \quad i \neq j \in \{1, \ldots, t\}$$

Since λ may not be an integer, the aim is to minimize the number of repeated co-occurrences of treatment pairs within the same block and look for the following condition:

$$\max_{i \neq j} C_{i,j} \leq \lceil \lambda \rceil$$

(C3) Session-Wise Balance (if applicable)

Let $\mathcal{B}_s \subset \mathcal{B}$ be the set of blocks in session s with $|\mathcal{B}_s| = m = b/s$. Then:

$$\sum_{B \in \mathcal{B}_s} A_{T,B} = \frac{r}{s} \quad \forall T \in \mathcal{T},\ \forall S \in \mathcal{S} \text{ provided that: } \frac{r}{s} \in \mathbb{N}$$

This condition ensures that each treatment appears equally often in each session, thereby maintaining orthogonality between treatment effects and session effects.

A secondary goal is to maximize D-optimality, measured by the determinant of the pseudo-regressor matrix associated with the design. Condition (*C2*) thus aligns with the classical definition of a packing design. The main distinction lies in the optimization goal: while packing designs are combinatorially feasible configurations, the proposed heuristic simultaneously optimizes for statistical efficiency, producing near-optimal quasi-BIBDs that satisfy (*C2*) and (*C3*) while maximizing the determinant of the information matrix.

Table 3 Algorithm Execution Log of Youden Square. ($t = b = 5$, $k = r = 4$, $\lambda = 3$)

Iteration	i	j	T_i	T_j	k	Determinant	R_λ	ℓ	f_ℓ	Shuffle
0	1	1	A	A	1/1	4.09E+09	8	4	2	0
1	1	5	A	B	1/1	7.91E+09	4	4	1	0
2	2	14	E	D	2/2	8.10E+09	0	3	7	0
3	8	20	B	D	4/4	8.10E+09	0	3	7	0
4	8	20	D	B	4/4	8.10E+09	0	3	7	0
5	2	10	D	E	2/2	2.63E+09	9	5	1	1
6	11	15	E	B	3/3	2.34E+09	5	5	1	1
7	4	8	C	B	4/4	3.03E+09	4	4	1	1
8	6	18	C	A	2/2	6.01E+09	0	3	5	1
9	1	13	C	E	1/1	6.82E+09	4	4	1	1
10	3	19	C	A	3/3	6.81E+09	0	3	8	1
11	4	20	B	E	4/4	4.44E+09	0	3	8	1
12	4	8	E	C	4/4	6.81E+09	0	3	8	1
13	4	8	C	E	4/4	4.44E+09	0	3	8	1
14	5	9	C	E	1/1	1.08E+10	4	4	1	2
15	1	17	B	D	1/1	1.27E+10	0	3	10	2

```
Require: $t$, $k$, $b$, $r$, $I_{\max}$
if session = TRUE then
Require: $s$
    $m \leftarrow t/s$
else
    $s \leftarrow b$, $m \leftarrow 1$
end if
Ensure: Optimal (quasi-)BIBD $M$ satisfying ($C1$) and approximating ($C2$)
if session = TRUE then
Ensure: satisfying ($C3$)
end if
$n \leftarrow b \times k$
Initialize matrix $M^0_{(n\times 3)}$
Set $M^0[:,1]$ as block labels, each repeated $k$ times;
Set $M^0[:,2]$ as session labels $b$ or $s$, repeated $k$ times each
for $j = 1$ to $s$ do
    Set $M^0[which(M[,2] = j), 3]$ as a random permutation of treatments    ▷ ($C1$)
end for
$M \leftarrow M^0$
Compute $R_\lambda$, $D$, $\ell$ and $f\ell$    ▷ ($C2$)
while $I \le I_{\max}$ do
    $M_{\text{prev}} \leftarrow M$
    for $i = 1$ to $n$ do
        for $j = i + 1$ to $n$ do
            if $M[i,3] \neq M[j,3]$ and ($M[i,1] = M[j,1]$ or $M[i,2] = M[j,2]$) then
                Create $M'$ by swapping $M[i,3]$ and $M[j,3]$
                Compute $R'_\lambda$, $D'$, $\ell'$, $f'_\ell$
                if $R'_\lambda < R_\lambda$ then
                    Accept swap
                else if $R'_\lambda = R_\lambda$ and $D' > D$ then
                    Accept swap
                else if $R'_\lambda = R_\lambda$ and $D' = D$ and $\ell' < \ell$ then
                    Accept swap
                else if $R'_\lambda = R_\lambda$ and $D' = D$ and $\ell' = \ell$ and $f'_\ell < f_\ell$ then
                    Accept swap
                end if
                if swap accepted then
                    $M \leftarrow M'$, $R_\lambda \leftarrow R'_\lambda$, $D \leftarrow D'$, $\ell \leftarrow \ell'$, $f_\ell \leftarrow f'_\ell$
                    break to next $i$
                end if
            end if
        end for
    end for
    if $M = M_{\text{prev}}$ then
        if shuffle = TRUE then
            Perform constrained shuffle to $M$
            $M^0 \leftarrow M$
            continue    ▷ restart search from new initial design
        else
            break    ▷ terminate: no improving swap found
        end if
    end if
    $I \leftarrow I + 1$
end while
return $M$, $R_\lambda$, $D$, $\ell$, $f_\ell$
```

Algorithm 1 Pseudo-code of the algorithm for constructing quasi-BIBD

Two metrics are computed for each configuration: R_λ and D. The former corresponds to the weighted count of treatment pairs exceeding the allowed frequency threshold, emphasising violations above λ; the latter refers to the determinant of the pseudo-regressor matrix constructed from the treatment assignments, representing the D-optimality of the design. Formally, given $c_k = k\,(k-1)/2$ be the number of unordered pairs within a single block of size k, $f = (f_1, \ldots, f_{c_k})$ be the vector of the frequency of distinct unordered treatment pairs that co-occur exactly $\ell = 0, 1, \ldots, c_k$ times across all blocks, then:

$$R_\lambda = \sum_{\ell=\lceil\lambda\rceil+1}^{c_k} \ell \cdot f_\ell \qquad D = \det \left(X^\top X\right)$$

where X is the pseudo-regression matrix. The algorithm proceeds by iteratively improving the configuration of the design through local exchanges. In cases where R_λ and D provide identical values across configurations, the optimisation continues by favouring designs that reduce first the maximum within-block co-occurrence of any treatment pair (ℓ) and then the corresponding frequency (f_ℓ). For illustrative purpose, consider the Youden Square reported in Table 1. The corresponding values of the selected metrics for that configuration, are shown in the last row of Table 3).

Two treatments T_i and T_j may be swapped only if:

- They belong to the same block (to preserve block size k), or
- They belong to the same session (to preserve session-wise balance)

After each candidate exchange, recompute R_λ, D and f: the exchange is accepted if it results in a decrease in R_λ or, secondarily, in an increase in D. This guarantees that both treatment balance and session balance are preserved throughout the optimization. Among competing designs satisfying this criterion, the one with the highest D-optimality ensures both structural robustness and inferential efficiency. When R_λ and D remain unchanged, the swap is accepted if it produces a reduction in ℓ or, when ℓ is unchanged, a reduction in f_ℓ. The search proceeds until a complete round of iterations is made without accepting any exchanges. The algorithm compares the current configuration M with the one obtained in the previous iteration. If the same configuration is encountered twice consecutively, the optimisation is considered stagnated. At this point, the procedure may either terminate or, alternatively, perform a constrained shuffle that preserves block size k (and session structure s, when present). In this case, the search restarts from the first iteration using the reshuffled configuration. Such a shuffle allows the search to escape local optima while maintaining the required constraints. The procedure continues until the predefined global iteration limit is reached ($I_{\max}$), which serves as the final stopping criterion.

Due to the inherent characteristic of heuristic search methods, which typically converge to local rather than global optima depending on the initial solution, it is recommended to introduce variability in the starting configurations to increase the chances of reaching better solutions. This can be achieved by random perturbations of the initial constrained assignment. Such permutations alter the starting point of the search without affecting the first-order balance condition, which remains guaranteed by the initial construction of the design.

The algorithm produces a structured output log that records the evolution of each iteration and the corresponding evaluation metrics. This log supports transparent assessment of the search process and facilitates comparisons across alternative configurations. Each row of the log corresponds to a single iteration of the algorithm and reports: (i) the swap details (i, j, T_i, T_j); (ii) key diagnostic quantities such as the determinant of the information matrix *D*, the weighted count of violations R_λ, the maximum co-occurrence multiplicity observed among all unique pairs ℓ and the number of distinct treatment pairs that achieve the maximum multiplicity f_ℓ; and (iii) an auxiliary indicator used to monitor algorithmic stability, including shuffle occurrences. Together, these elements provide a comprehensive trace of the algorithm's behaviour and enable a detailed examination of convergence patterns, design quality, and potential sources of sub-optimality.

To illustrate the structure and interpretation of the log, Table 3 collects the complete output generated for the Youden Square example (see Table 1). The resulting log captures the progression toward a stable configuration and highlights how the diagnostic metrics reflect improvements in balance and information efficiency. Each row corresponds to a single attempted swap between two treatments in the long-format design, and the columns summarize all quantities used to evaluate and update the design. The columns *i* and *j* indicate the row indices of the two treatments being exchanged, T_i and T_j. The column *k* identifies the factor column in which the swap takes place. Each entry (e.g. "1/1") indicates that both elements exchanged belong to the same factor column, ensuring that every swap is performed within that column only. This preserves all structural constraints of the design and, when sessions are used, an additional column *s* is provided: it follows the same notation (e.g. "1/1" meaning that the swap also occurs within the same session), thereby guaranteeing that session assignments are likewise preserved. The column Determinant reports the value of *D* computed after each swap; although not the primary optimisation target, its steady increase across iterations reflects the progressive improvement of the information matrix. The column R_λ represents the main objective function, capturing the weighted violations of (*C2*); its decrease shows that irregular treatment co-occurrences are being effectively eliminated. The quantities ℓ and f_ℓ summarise the distribution of pairwise co-occurrence counts. Together, these indicators monitor the balance between treatment combinations during the optimization trajectory. The final column (*Shuffle*) identifies iterations in which the algorithm enters a stagnation phase. When no improving swaps are available, a constrained resampling step is triggered, reassigning treatments within the same column (or within the same session, if present) to escape local optima while maintaining all design constraints.

It is worth noticing that the algorithm continues iterating even though the stopping condition would normally be met ($R_\lambda = 0$): in this illustrative example only, the initial constraint on block-wise balance was intentionally relaxed for explanatory purposes. As a consequence, some blocks contain repeated treatments (for example, see iteration number 2 where treatment A appears three times in Block 1 in the corresponding design), a situation that is not allowed in the problem formulation adopted in the manuscript. Because this constraint is violated in the illustrative setup, the algorithm does not stop when the usual criteria are satisfied. The sequence of iterations is shown precisely to illustrate how convergence is achieved once all structural constraints of the design are respected. Overall, the algorithm converges in a small number of iterations, achieves a clear increase in the determinant, reduces the value of the primary objective R_λ, and maintains all structural constraints by restricting swaps to occur within fixed factor levels (and fixed sessions, when applicable).

4 Two illustrative examples

The practical value of balanced incomplete block designs, even in approximate form, becomes evident in experimental scenarios where a limited number of evaluators or environmental conditions prevent the full execution of a complete factorial design. In such settings, the simultaneous presence of first-order balance (*C1*) and approximate second-order balance (*C2*) conditions becomes central to mitigating potential confounding effects. Two examples illustrating this strategy are presented in the following sections.

The experimental designs resulting from the optimization process for each case study are provided in the Appendix, together with the corresponding algorithm execution log and the graphical representation of the associated treatment–concurrence graph.

4.1 Example 1: product evaluation under resource constraints

In a marketing or production study aimed at understanding consumer perception of a product, researchers examined the effect of eight binary factors. For example, the proposed methodology was applied to an industrial case involving the optimization of a coffee blend formulation under economic constraints. This need arose due to recent fluctuations in the market price of raw coffee materials, prompting a reformulation effort aimed at minimizing quality loss while reducing production costs. In addition to optimizing the blend composition, the experimental design incorporated two key roasting process parameters—time and burner power—selected from practical levels commonly used in industrial roasting. The aim was to test a range of feasible combinations that reflect standard operational conditions without resorting to a full factorial design that would have required 2^8 runs. However, using a standard fractional factorial approach, a reduced design of 20 runs could be employed, effectively representing 20 distinct experimental treatments ($2^{8-4} + 4$ design, i.e. a fractional two-level design with 16 runs augmented with 4 center points). Then, the experiment aimed to identify a new optimal blend of roasted coffee by testing t=20 different formulations, each characterized by the specific composition of six different green coffee types and two roasting process parameters (time and burner power).

Given the limited availability of experts, only $k = 4$ evaluators could be involved. Each treatment was to be assessed once by each evaluator, yielding $r = 4$ replicates per treatment, for a total of $n = tr = bk = 80$ evaluations. Since t and b are equal, the design can, in principle, be structured as a Youden square. This arrangement ensures that each treatment appears once per block, and that every evaluator assesses all treatments. As a result, the design incorporates two blocking factors: the evaluator (represented by columns) and the evaluation occasion (represented by blocks), similarly to a Latin square, where both row and column effects are balanced.

It is important to highlight that the BIBD structure plays a key role in achieving constrained randomization of the run distribution. Consider what might occur if specific treatment pairs repeatedly appeared together in the same block. Those blocks should be affected by some external disturbance, the interaction between factors would be systematically biased, potentially leading to a spurious significance that does not reflect the actual treatment effects but rather a flaw in randomization.

The design structure corresponds to a BIBD with $t = b = 20, k = 4$, and $r = 4$, where each block corresponds to a group of 4 different treatments simultaneously assessed by different evaluators. While first-order balance (each treatment evaluated equally often) was ensured by construction, exact second-order balance could not be achieved. In fact, the expected value of the pairwise concurrence parameter,

$$\lambda = \frac{20\begin{pmatrix} 4 \\ 2 \end{pmatrix}}{\begin{pmatrix} 20 \\ 2 \end{pmatrix}} = \frac{20 \cdot 6}{190} = \frac{12}{19},$$

is not an integer, implying that no exact BIBD exists in this configuration. Note that $\lambda < 1$, thus, an approximate second-order balanced design, where most treatment pairs co-occur at most once, was the most practical solution.

The structure of the problem permits the blocks to be organized into $s = 4$ sessions of $m = 5$ blocks each. Each session contains all 20 treatments exactly once, achieving session-wise balance and allowing simultaneous execution by the evaluators over separate evaluation rounds. This structure is analogous to a Youden square, wherein treatments are equally distributed across evaluators (columns) and sessions (rows), and blocks are nested within sessions.

This organization is especially valuable in maintaining robustness to time-related biases or evaluator fatigue. If, for instance, treatment pairs were systematically overrepresented in a specific session and that session were compromised due to a systematic error (e.g. environmental disturbance or evaluator drift), the resulting confounding could produce spurious interactions. Randomizing the order of block presentation within each session further mitigates this risk.

4.2 Example 2: robust design with control and noise factors

The second case study illustrates the application of the proposed algorithm in the context of robust design, involving four continuous Control Factors (CF) and two categorical Noise Factors (NF) at three levels (Low, Medium, High). Taguchi's classical approach employs a *Product Array*, combining separate designs for CF and NF through a Cartesian product (Taguchi 1987). CF combinations, named *prototypes* as they represent technological designs which are tested under various environmental conditions, thus evaluated across all NF settings. This approach has been criticized because it allows estimation of all possible CF$\times$NF interaction terms, including higher-order ones that are likely irrelevant. While comprehensive, this strategy often results in excessive runs becoming resource-intensive.

As an alternative, Box and Jones (1992a) proposed the *Combined Array* methodology, which estimates only pairwise (two-factor) interactions, assuming higher-order interactions are negligible. This model treats all factors simultaneously in a single design, and the response model takes the form:

$$y = \beta_0 + \boldsymbol{x}^T\boldsymbol{\beta} + \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x} + \boldsymbol{z}^T\gamma + \boldsymbol{z}^T\boldsymbol{\Gamma}\boldsymbol{z} + \boldsymbol{z}^T\boldsymbol{\Delta}\boldsymbol{x}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{B}$ are the coefficients of the first and second order of CF, γ and $\boldsymbol{\Gamma}$ of NF, and $\boldsymbol{\Delta}$ of the CF$\times$NF interactions. In the general case, in which the quadratic components of the NF are not absent, the authors show that the objective function to be minimized is the following:

$$\mathcal{L}(\boldsymbol{x}) = [\tau - (\beta_0 + \boldsymbol{x}^T\boldsymbol{\beta} + \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x} + \frac{1}{3}\ \mathrm{tr}\ (\boldsymbol{\Gamma}))]^2 + \frac{1}{3}(\gamma + \boldsymbol{\Delta}\boldsymbol{x})^T(\gamma + \boldsymbol{\Delta}\boldsymbol{x})$$

where τ is the target value, and tr( ) denotes the matrix trace. Therefore, the coefficients of CF, interactions CF$\times$NF, and only linear and quadratic pure of NF have to be estimated, but for the latter only in a joint way. At this aim, a Central Composite Design (CCD) can be used. Observe that, as the quadratic component of NF should be evaluated as a sum, the point related to the "star" ($-\alpha$, $+\alpha$) of the $\boldsymbol{z}$, can be omitted. This allows the use of three-level categorical noise factor, not necessarily continuous.

Suppose we have 4 CF and 2 NF. A standard CCD with 6 factors involves 54 runs. The 4 runs for the pure quadratic components of NF can be removed, reducing the design to 50 runs. In practice, the prototypes are 16 at (-1,$+1$) levels associated at 2 environmental condition (-1,$+1$), and 8 at ($-\alpha$, $+\alpha$) levels associated at 0 level environmental conditions. Moreover, 10 replicates are foreseen at 0 level for all factors (central points).

Since the trials must be independent, separate prototypes should be prepared for each experimental condition. Furthermore, each environmental condition should ideally be prepared independently as well. This must be accomplished also for the replicated central points that must be prepared independently of each other. This allows to evaluate the replicability of performance of prototypes, while using the same prototype several times assures only to observe the reproducibility.

This approach would require an enormous amount of resources. In practice, it is more efficient to reconsider the product array strategy, where a single prototype, assembled once, is subjected to all environmental conditions. Therefore, prototypes should be treated as treatments within a split-plot design framework and should be experimented multiple times, varying the environmental condition. Thus, Combined Arrays provide an efficient strategy for separately designing the control and noise factors, while the Product Array approach allows for substantial savings in experimental effort.

So, a CCD with 4 CF and 30 prototype combinations corresponds to 16 prototypes at (-1,$+1$) levels, 8 at ($-\alpha$, $+\alpha$) and 1 at (0, 0) replicated 6 times. The latter 6 prototypes assure the possibility to separate the reproducibility and the replicability effect, as before precised. In this case, each prototype can be submitted at the 5 environmental condition: four at $(-1,+1)$ levels and one at (0,0) level, leading to 150 total runs (30×5). Though this triples the number of runs compared to the combined design (150 runs instead of 50), the number of CF settings (prototypes) are reduced (30 instead of 50).

As in the previous example, these prototypes now play the role of treatments, and their evaluation order within environmental configurations (i.e., sessions) must be carefully arranged using the proposed algorithm to preserve balance and minimize bias.

As an illustrative example, let refer to Box and Jones (1992b), where a cake recipe depends on 4 factors (ingredients), and an oven that can be set according to 2 variables (time and temperature). The experimental design consisted of a Central Composite Design (CCD) for the CF, resulting in 30 unique treatment combinations (16 factorial points, 8 axial points, and 6 center points). Each treatment was evaluated under 5 different environmental conditions (e.g., oven temperature and time) representing the NF through a 2-level full factorial design plus a center point. Each block thus consisted of $k = 5$ prototype evaluations conducted concurrently under different environmental settings. Each of the $t = 30$ treatment combinations was replicated $r = 5$ times, yielding $n = tr = bk = 150$ total runs.

Formally, the structure corresponds to a BIBD with $t = b = 30$, $k = 5$, and $r = 5$. However, the ideal pairwise replication value

$$\lambda = \frac{30\begin{pmatrix} 5 \\ 2 \end{pmatrix}}{\begin{pmatrix} 30 \\ 2 \end{pmatrix}} = \frac{30 \cdot 10}{435} = \frac{20}{29}$$

is not an integer, precluding the existence of an exact BIBD. As in the previous example, an approximate solution was thus constructed to ensure that most treatment pairs appeared together at most once per block, minimizing redundancy while preserving estimability of model parameters.

Given that $k = 5$ divides evenly into $t = 30$, the design was organized into $s = 5$ sessions, each comprising $m = 6$ blocks. This configuration allowed for uniform treatment representation across sessions and ensured that block-level randomization could be constrained within session-level execution, enhancing experimental control over environmental variation. As with the first example, randomization of treatment order within each environmental condition was essential to prevent systematic bias from being aligned with specific CF settings. The session-based layout, analogous again to a Youden square, ensured balanced comparisons while minimizing confounding.

These two examples highlight how approximate BIBD structures, organized across sessions and integrated with constrained randomization, can address practical limitations in experimental execution while preserving key statistical properties. In both cases, exact second-order balance was unattainable, yet careful design enabled near-optimal configurations suitable for reliable estimation and inference.

5 Conclusion

This work addresses a gap in experimental design practice concerning the construction of balanced incomplete block structures that preserve first-order balance, approximate second-order balance, and accommodate flexible session arrangements. The designs considered here are quasi-BIBDs produced through a heuristic optimisation procedure that iteratively improves pairwise balance while maintaining the structural constraints of the design. Unlike classical resolvable or nested BIBDs, which rely on the existence of an exact combinatorial configuration and hence on an integer concurrence parameter λ, the present framework explicitly admits non-integer λ and does not require an exact BIBD to exist.

In cases where exact BIBDs do not exist, our approach offers a practical and effective alternative by identifying near-optimal configurations that minimize treatment-pair redundancy and maximize D-optimality. This makes the approach applicable in settings where combinatorial feasibility is unattainable but balanced allocation remains essential. Within this structure, sessions serve as a blocking factor rather than as exact replicates of the block system. Their purpose is to distribute treatments evenly across the second factor, namely the position within blocks. Such positional balance is not enforced session by session but emerges only when the design is considered in its entirety. This provides the flexibility needed to accommodate logistical constraints, while still ensuring that treatment representation remains orthogonal to session-level variability. Notably, the resulting designs also satisfy global connectedness, guaranteeing that all treatment contrasts remain estimable and that no loss of degrees of freedom arises from disconnected treatment–association structures.

The resulting methodology is particularly relevant in contemporary experimental scenarios, such as consumer studies, sensory evaluation, and robust industrial experimentation, where the number of conditions to be assessed often exceeds what can be simultaneously tested in full factorial form. The proposed algorithm ensures designs with a more balanced distribution of treatment combinations across evaluators and time blocks, thereby reducing the likelihood of confounding or biased contrasts arising from poor randomisation or unequal exposure.

Some limitations remain, as the heuristic nature of the method does not guarantee global optimality and its performance may depend on the quality of the initial random configuration or on the number of iterations executed. Future developments could aim to formalise the second-order balance requirement as a soft constraint or penalty term within the optimization objective, enabling a more explicit trade-off between D-optimality and pairwise uniformity.

Appendix A collects the designs obtained for each example, together with their evaluation in terms of treatment–pair recurrences within blocks. The section also reports the complete algorithm execution log and the graphical representation of the corresponding treatment–concurrence graph.

Appendix A The proposed designs

A.1 Example 1 (without session)

See Fig. 2, Tables 4 and 5.

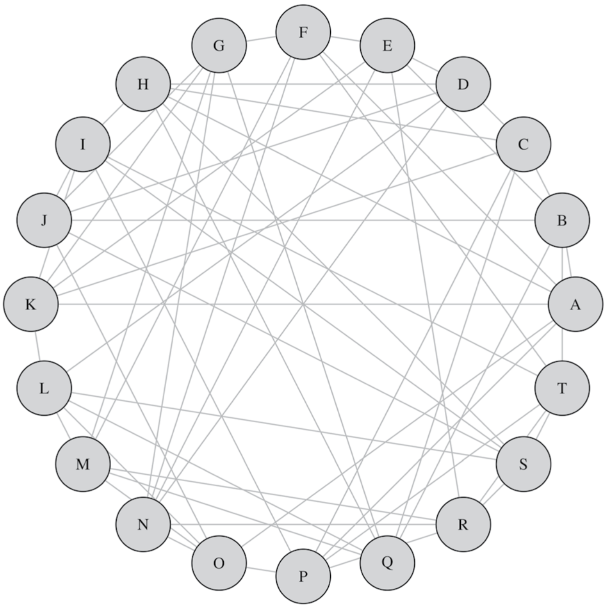

Fig. 2 Connected design in Table 5

Table 4 Algorithm Execution Log of Example 1 ($t = b = 20$, $k = r = 4$, $\lambda = 0.63$)

Iteration	i	j	T_i	T_j	k	Determinant	R_λ	ℓ	f_ℓ	Shuffle
0	1	1	D	D	1/1	1.08E+29	65	3	5	0
1	13	25	N	O	1/1	1.76E+29	56	3	4	0
2	17	45	A	E	1/1	2.36E+29	47	3	3	0
3	24	80	S	C	4/4	3.20E+29	39	3	1	0
4	68	72	R	I	4/4	4.79E+29	31	3	1	0
5	10	30	K	J	2/2	5.16E+29	23	3	1	0
6	43	71	P	I	3/3	5.49E+29	17	3	1	0
7	36	60	F	E	4/4	5.77E+29	11	3	1	0
8	42	50	M	B	2/2	6.08E+29	8	2	4	0
9	46	54	E	R	2/2	6.22E+29	6	2	3	0
10	3	75	G	M	3/3	6.32E+29	4	2	2	0
11	3	15	M	F	3/3	6.38E+29	2	2	1	0
12	28	68	O	I	4/4	6.42E+29	2	2	1	0
13	28	56	I	J	4/4	6.51E+29	0	1	200	0

Table 5 Example 1. ($t = b = 20$, $k = r = 4$, $\lambda = 0.63$) Proposed design (table on the top-left). Treatment pairs in the same block (table on the top-right). Number of occurrences of pairs in the same block (table on the bottom)

Blocks	I	II	III	IV
1	A	B	C	D
2	B	E	F	G
3	C	H	I	J
4	D	J	G	K
5	E	D	L	M
6	F	N	D	H
7	G	M	O	P
8	H	Q	M	R
9	I	K	E	N
10	J	S	R	E
11	K	C	P	T
12	L	O	J	B
13	M	F	T	I
14	N	G	Q	C
15	O	A	H	S
16	P	I	S	L
17	Q	L	K	A
18	R	P	A	F
19	S	T	B	Q
20	T	R	N	O

AB	AC	AD	BC	BD	CD
BE	BF	BG	EF	EG	FG
CH	CI	CJ	HI	HJ	IJ
DJ	DG	DK	GJ	JK	GK
DE	EL	EM	DL	DM	LM
FN	DF	FH	DN	HN	DH
GM	GO	GP	MO	MP	OP
HQ	HM	HR	MQ	QR	MR
IK	EI	IN	EK	KN	EN
JS	JR	EJ	RS	ES	ER
CK	KP	KT	CP	CT	PT
LO	JL	BL	JO	BO	BJ
FM	MT	IM	FT	FI	IT
GN	NQ	CN	GQ	CG	CQ
AO	HO	OS	AH	AS	HS
IP	PS	LP	IS	IL	LS
LQ	KQ	AQ	KL	AL	AK
PR	AR	FR	AP	FP	AF
ST	BS	QS	BT	QT	BQ
RT	NT	OT	NR	OR	NO

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	1	1	1	0	1	0	1	0	0	1	1	0	0	1	1	1	1	1	0
B		1	1	1	1	1	0	0	1	0	1	0	0	1	0	1	0	1	1
C			1	0	0	1	1	1	1	1	0	0	1	0	1	1	0	0	1
D				1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0
E					1	1	0	1	1	1	1	1	1	0	0	0	1	1	0
F						1	1	1	0	0	0	1	1	0	1	0	1	0	1
G							0	0	1	1	0	1	1	1	1	1	0	0	0
H								1	1	0	0	1	1	1	0	1	1	1	0
I									1	1	1	1	1	0	1	0	0	1	1
J										1	1	0	0	1	0	0	1	1	0
K											1	0	1	0	1	1	0	0	1
L												1	0	1	1	1	0	1	0
M													0	1	1	1	1	0	1
N														1	0	1	1	0	1
O															1	0	1	1	1
P																0	1	1	1
Q																	1	1	1
R																		1	1
S																			1

A.2 Example 1 (with session)

See Fig. 3, Tables 6 and 7.

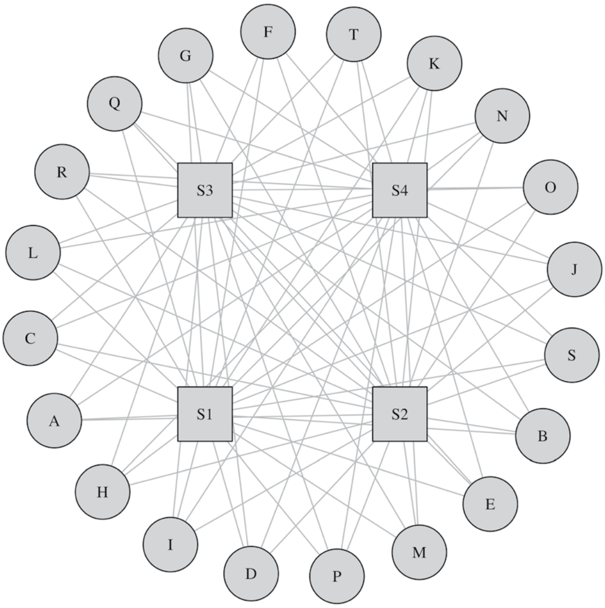

Fig. 3 Connected design in Table 7

Table 6 Algorithm Execution Log of Example 1 ($t = b = 20$, $k = r = 4$, $s = 4$, $\lambda = 0.63$)

Iteration	i	j	T_i	T_j	k	s	Determinant	R_λ	ℓ	f_ℓ	Shuffle
0	1	1	D	D	1/1	1/1	8.82E+33	55	3	7	0
1	43	75	E	B	3/3	3/3	5.00E+33	62	3	4	0
2	25	73	N	L	1/1	3/3	5.89E+33	53	3	3	0
3	15	79	J	H	3/3	4/4	6.76E+33	44	3	2	0
4	17	65	E	C	1/1	1/1	7.18E+33	38	3	2	0
5	30	46	M	O	2/2	4/4	7.64E+33	32	3	2	0
6	42	58	P	Q	2/2	3/3	8.05E+33	29	3	1	0
7	4	52	T	S	4/4	1/1	8.46E+33	26	2	13	0
8	40	72	K	M	4/4	2/2	8.59E+33	24	2	12	0
9	35	51	N	M	3/3	1/1	8.69E+33	22	2	11	0
10	19	51	L	N	3/3	1/1	8.83E+33	20	2	10	0
11	32	80	A	C	4/4	4/4	9.06E+33	16	2	8	0
12	5	69	G	H	1/1	2/2	9.21E+33	14	2	7	0
13	47	63	F	G	3/3	4/4	9.35E+33	12	2	6	0
14	32	64	C	B	4/4	4/4	9.41E+33	10	2	5	0
15	30	62	O	L	2/2	4/4	9.38E+33	12	2	6	0
16	35	67	M	O	3/3	1/1	9.37E+33	9	3	1	0
17	29	77	R	Q	1/1	4/4	9.36E+33	9	3	1	0
18	32	48	B	D	4/4	4/4	9.37E+33	9	3	1	0
19	29	77	Q	R	1/1	4/4	9.36E+33	9	3	1	0
20	29	77	R	Q	1/1	4/4	9.37E+33	9	3	1	0
21	22	54	D	A	2/2	2/2	6.45E+33	45	3	3	1
22	1	49	B	E	1/1	1/1	6.84E+33	38	3	4	1
23	18	66	I	F	2/2	1/1	7.26E+33	32	3	4	1
24	46	62	N	M	2/2	4/4	7.62E+33	29	3	3	1
25	15	31	H	F	3/3	4/4	7.99E+33	26	3	2	1
26	35	67	N	K	3/3	1/1	8.43E+33	23	3	1	1
27	8	24	K	M	4/4	2/2	8.58E+33	21	3	1	1
28	32	48	E	C	4/4	4/4	8.57E+33	18	3	2	1
29	37	69	H	G	1/1	2/2	8.82E+33	17	3	1	1
30	24	72	K	L	4/4	2/2	9.42E+33	12	2	6	1
31	47	79	G	I	3/3	4/4	9.75E+33	8	2	4	1
32	46	78	M	L	2/2	4/4	1.04E+34	0	1	200	1

Table 7 Example 1. ($t = b = 20$, $k = r = 4$, $s = 4$, $\lambda = 0.63$) Proposed design taking into account the *sessions* (table on the top-left). Treatment pairs in the same block (table on the top-right). Number of occurrences of pairs in the same block (table on the bottom)

Blocks	I	II	III	IV	S
1	A	F	K	P	1
2	B	G	L	Q	1
3	C	H	M	R	1
4	D	I	N	S	1
5	E	J	O	T	1
6	F	B	R	N	2
7	G	C	S	O	2
8	H	D	T	K	2
9	I	E	P	L	2
10	J	A	Q	M	2
11	K	Q	C	I	3
12	L	R	D	J	3
13	M	S	E	F	3
14	N	T	A	G	3
15	O	P	B	H	3
16	P	M	G	D	4
17	Q	N	H	E	4
18	R	O	I	A	4
19	S	K	J	B	4
20	T	L	F	C	4

AF	AK	AP	FK	FP	KP
BG	BL	BQ	GL	GQ	LQ
CH	CM	CR	HM	HR	MR
DI	DN	DS	IN	IS	NS
EJ	EO	ET	JO	JT	OT
BF	FR	FN	BR	BN	NR
CG	GS	GO	CS	CO	OS
DH	HT	HK	DT	DK	KT
EI	IP	IL	EP	EL	LP
AJ	JQ	JM	AQ	AM	MQ
KQ	CK	IK	CQ	IQ	CI
LR	DL	JL	DR	JR	DJ
MS	EM	FM	ES	FS	EF
NT	AN	GN	AT	GT	AG
OP	BO	HO	BP	HP	BH
MP	GP	DP	GM	DM	DG
NQ	HQ	EQ	HN	EN	EH
OR	IR	AR	IO	AO	AI
KS	JS	BS	JK	BK	BJ
LT	FT	CT	FL	CL	CF

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A	0	0	0	0	1	1	0	1	1	1	0	1	1	1	1	1	1	0	1
B		0	0	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	0
C			0	0	1	1	1	1	0	1	1	1	0	1	0	1	1	1	1
D				0	0	1	1	1	1	1	1	1	1	0	1	0	1	1	1
E					1	0	1	1	1	0	1	1	1	1	1	1	0	1	1
F						0	0	0	0	1	1	1	1	0	1	0	1	1	1
G							0	0	0	0	1	1	1	1	1	1	0	1	1
H								0	0	1	0	1	1	1	1	1	1	0	1
I									0	1	1	0	1	1	1	1	1	1	0
J										1	1	1	0	1	0	1	1	1	1
K											0	0	0	0	1	1	0	1	1
L												0	0	0	1	1	1	0	1
M													0	0	1	1	1	1	0
N														0	0	1	1	1	1
O															1	0	1	1	1
P																0	0	0	0
Q																	0	0	0
R																		0	0
S																			0

A.3 Example 2

See Fig. 4, Table 8 and 9.

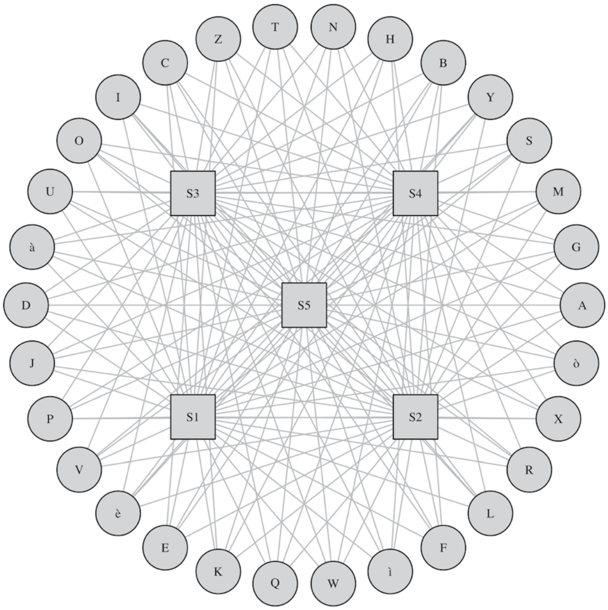

Fig. 4 Connected design in Table 9

Table 8 Algorithm Execution Log of Example 2 ($t = b = 30$, $k = r = 5$, $s = 5$, $\lambda = 0.69$)

Iteration	i	j	T_i	T_j	k	s	Determinant	R_λ	ℓ	f_ℓ	Shuffle
0	1	1	A	A	1/1	1/1	8.41E+55	82	2	41	0
1	24	49	O	R	4/4	5/5	8.75E+55	70	2	35	0
2	47	122	V	X	2/2	5/5	9.02E+55	60	2	30	0
3	75	125	A	F	5/5	5/5	9.23E+55	52	2	26	0
4	99	149	N	M	4/4	5/5	9.40E+55	46	2	23	0
5	25	100	D	E	5/5	5/5	9.60E+55	38	2	19	0
6	73	148	H	K	3/3	5/5	9.67E+55	34	2	17	0
7	46	71	Y	è	1/1	5/5	9.82E+55	28	2	14	0
8	49	74	O	Q	4/4	5/5	1.00E+56	20	2	10	0
9	124	149	P	N	4/4	5/5	1.00E+56	20	2	10	0
10	74	124	O	N	4/4	5/5	1.00E+56	20	2	10	0
11	74	124	N	O	4/4	5/5	1.00E+56	20	2	10	0
12	24	124	O	P	4/4	5/5	8.52E+55	80	2	40	1
13	48	123	K	J	3/3	5/5	8.75E+55	70	2	35	1
14	73	98	H	G	3/3	5/5	8.96E+55	62	2	31	1
15	49	74	Q	M	4/4	5/5	9.18E+55	54	2	27	1
16	47	97	X	T	2/2	5/5	9.33E+55	48	2	24	1
17	100	150	D	C	5/5	5/5	9.48E+55	42	2	21	1
18	49	149	M	N	4/4	5/5	9.60E+55	38	2	19	1
19	47	147	T	U	2/2	5/5	9.70E+55	34	2	17	1
20	46	146	ò	è	1/1	5/5	9.80E+55	30	2	15	1
21	72	122	V	S	2/2	5/5	9.89E+55	26	2	13	1
22	72	97	S	X	2/2	5/5	1.02E+56	16	2	8	1
23	21	71	Y	à	1/1	5/5	1.03E+56	12	2	6	1
24	46	71	è	Y	1/1	5/5	1.06E+56	0	1	450	1

Table 9 Example 2. ($t = b = 30$, $k = r = 5$, $s = 5$, $\lambda = 0.69$) Proposed design taking into account the *sessions* (table on the top-left). Treatment pairs in the same block (table on the top-right). Number of occurrences of pairs in the same block (table on the bottom)

Blocks	I	II	III	IV	V	s
1	A	G	M	S	Y	1
2	B	H	N	T	Z	1
3	C	I	O	U	à	1
4	D	J	P	V	è	1
5	E	K	Q	W	ì	1
6	F	L	R	X	ò	1
7	G	N	C	ò	V	2
8	H	O	D	Y	W	2
9	I	P	E	Z	X	2
10	J	Q	F	à	S	2
11	K	R	A	è	T	2
12	L	M	B	ì	U	2
13	M	Z	W	C	J	3
14	N	à	X	D	K	3
15	O	è	S	E	L	3
16	P	ì	T	F	G	3
17	Q	ò	U	A	H	3
18	R	Y	V	B	I	3
19	S	B	ò	K	P	4
20	T	C	Y	L	Q	4
21	U	D	Z	G	R	4
22	V	E	à	H	M	4
23	W	F	è	I	N	4
24	X	A	ì	J	O	4
25	Y	U	J	N	E	5
26	Z	V	K	O	F	5
27	à	W	L	P	A	5
28	è	X	G	Q	B	5
29	ì	S	H	R	C	5
30	ò	T	I	M	D	5

AG	AM	AS	AY	GM	GS	GY	MS	MY	SY
BH	BN	BT	BZ	HN	HT	HZ	NT	NZ	TZ
àC	àI	àO	àU	CI	CO	CU	IO	IU	OU
Dè	DJ	DP	DV	èJ	èP	èV	JP	JV	PV
Eì	EK	EQ	EW	ìK	ìQ	ìW	KQ	KW	QW
FL	Fò	FR	FX	Lò	LR	LX	òR	òX	RX
CG	CN	Cò	CV	GN	Gò	GV	Nò	NV	òV
DH	DO	DW	DY	HO	HW	HY	OW	OY	WY
EI	EP	EX	EZ	IP	IX	IZ	PX	PZ	XZ
àF	àJ	àQ	àS	FJ	FQ	FS	JQ	JS	QS
Aè	AK	AR	AT	èK	èR	èT	KR	KT	RT
Bì	BL	BM	BU	ìL	ìM	ìU	LM	LU	MU
CJ	CM	CW	CZ	JM	JW	JZ	MW	MZ	WZ
àD	àK	àN	àX	DK	DN	DX	KN	KX	NX
Eè	EL	EO	ES	èL	èO	èS	LO	LS	OS
FG	Fì	FP	FT	Gì	GP	GT	ìP	ìT	PT
AH	Aò	AQ	AU	Hò	HQ	HU	òQ	òU	QU
BI	BR	BV	BY	IR	IV	IY	RV	RY	VY
BK	Bò	BP	BS	Kò	KP	KS	òP	òS	PS
CL	CQ	CT	CY	LQ	LT	LY	QT	QY	TY
DG	DR	DU	DZ	GR	GU	GZ	RU	RZ	UZ
àE	àH	àM	àV	EH	EM	EV	HM	HV	MV
èF	èI	èN	èW	FI	FN	FW	IN	IW	NW
Aì	AJ	AO	AX	ìJ	ìO	ìX	JO	JX	OX
EJ	EN	EU	EY	JN	JU	JY	NU	NY	UY
FK	FO	FV	FZ	KO	KV	KZ	OV	OZ	VZ
Aà	AL	AP	AW	àL	àP	àW	LP	LW	PW
Bè	BG	BQ	BX	èG	èQ	èX	GQ	GX	QX
CH	Cì	CR	CS	Hì	HR	HS	ìR	ìS	RS
DI	DM	Dò	DT	IM	Iò	IT	Mò	MT	òT

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	à	è	ì	ò
A	0	0	0	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1
B		0	0	0	0	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1
C			0	0	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1
D				0	0	1	1	1	1	1	0	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0	1
E					0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0
F						1	0	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1
G							0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1
H								0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1
I									0	0	0	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0	1
J										0	0	1	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0
K											0	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1
L												1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1
M													0	0	0	0	0	1	1	1	1	1	0	1	1	1	0	1	1
N														0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1
O															0	0	0	1	0	1	1	1	1	1	1	1	1	1	0
P																0	0	1	1	0	1	1	1	0	1	1	1	1	1
Q																	0	1	1	1	0	1	1	1	0	1	1	1	1
R																		1	1	1	1	0	1	1	1	0	1	1	1
S																			0	0	0	0	0	1	0	1	1	1	1
T																				0	0	0	0	1	1	0	1	1	1
U																					0	0	0	1	1	1	0	1	1
V																						0	0	1	1	1	1	0	1
W																							0	1	1	1	1	1	0
X																								0	1	1	1	1	1
Y																									0	0	0	0	0
Z																										0	0	0	0
à																											0	0	0
è																												0	0
ì																													0

Acknowledgements The authors would like to thank the Editor and two anonymous referees for their constructive comments, which highlighted several valuable aspects of the study and significantly improved the revised manuscript.

Author contributions All authors contributed to the study conception and design of the study. G.M. developed the methodology and carried out material preparation, algorithm implementation, and generation of computational results. All authors contributed to the drafting of the manuscript and to the revision of its earlier versions. All authors read and approved the final manuscript.

Funding Open access funding provided by Università degli Studi di Palermo within the CRUI-CARE Agreement. None.

Data availibility No data were used in the study to report.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no potential Conflict of interest with respect to the research, authorship, and/or publication of this article.

References

Abel RJR, Ge G, Yin J (2006) Resolvable and near-resolvable designs. In: Handbook of Combinatorial Designs, pp. 150–158. Chapman and Hall/CRC

Agashe NW, Varghese C, Harun M, Dalal A (2024) Tri-hierarchical incomplete block designs. Commn Stat Simulat Comput. https://doi.org/10.1080/03610918.2024.2339991

Benedict TJ, Fredrick O, Anthony K, Edward N (2023) Construction of non-symmetric balanced incomplete block design through combination of symmetric disjoint balanced incomplete block designs. Open J Stat. https://doi.org/10.4236/ojs.2023.136040

Box GEP, Hunter JS, Hunter WG (2005) Statistics for experimenters, 2nd edn. Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken, NJ

Box G, Jones S (1992) Designing products that are robust to the environment. Total Qual Manag 3(3):265–282

Box G, Jones S (1992) Split-plot designs for robust product experimentation. J Appl Stat 19(1):3–26

Buratti M, Kreher DL, Stinson DR (2024) Nestings of bibds with block size four. J Comb Des 32(12):715–743

Barone S, Lombardo A (2006) Balanced asymmetrical nearly orthogonal designs for first and second order effect estimation. J Appl Stat 33(4):373–386

Barone S, Lombardo A (2006) Construction and optimality of a special class of balanced designs. Qual Reliab Eng Int 22(5):507–515

Bose RC (1947) Proceedings of the Indian Science Congress vol. 34. Indian Science Congress Association and Asiatic Society of Bengal and Royal Asiatic Society of Bengal, Presidential Address

Chisaki S, Fuji-Hara R, Miyamoto N (2025) Optimality and constructions of spanning bipartite block designs. Metrika 88(2):247–265

Calinski T, Kageyama S (2012) Block Designs: A Randomization Approach: Volume II: Design vol. 170. Springer, New York

Draper NR, Smith H (1998) Applied Regression Analysis vol. 326, 3rd edn. John Wiley & Sons, New York

Eccleston J, Hedayat A (1974) On the theory of connected designs: characterization and optimality. Annal Stat 2(6):1238–1255

Giovagnoli A (1977) On the construction of factorial designs using abelian group theory. Rendiconti del Seminario Matematico della Università di Padova 58:195–206

Morgan JP, Preece DA, Rees DH (2001) Nested balanced incomplete block designs. Discret Math 231(1–3):351–389

Rueda DR, Cotta C, Fernández-Leiva AJ (2020) Memetic collaborative approaches for finding balanced incomplete block designs. Comput Oper Res 114:104804

Taguchi G (1987) System of experimental design; engineering methods to optimize quality and minimize costs

Team RC (2025) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria . R Foundation for Statistical Computing. https://www.R-project.org/

Wu CFJ, Hamada MS (2011) Experiments: planning, analysis, and optimization. John Wiley & Sons, Hoboken, NJ

Wynn H (1985) The Advanced Theory of Statistics, Vol. 3, Kendall, Sir Maurice, Stuart, A. and Ord, JK, High Wycombe. Wiley Online Library, Charles Griffin, 1983

Authors and Affiliations

Alberto Lombardo[1] · Giulia Marcon[1]

✉ Giulia Marcon
giulia.marcon@unipa.it

Alberto Lombardo
alberto.lombardo@unipa.it

[1] Department of Engineering, Università degli Studi di Palermo, Viale delle Scienze, 90128 Palermo, Italy