Statistica Sinica **33** (2023), 1903-1922
doi:https://doi.org/10.5705/ss.202021.0160

ON ORDERING PROBLEMS: A STATISTICAL APPROACH

Jianbin Chen[1], Xiaoxue Han[2], Dennis K.J. Lin[1],
Liuqing Yang[3] and Yongdao Zhou[3]

[1]*Purdue University,* [2]*Qufu Normal University and* [3]*Nankai University*

Abstract: In ordering problems, the goal is to find the optimal order. Each experimental run of an order problem is a permutation of m components. Because $m!$ is typically large, it is necessary to select a subset of the $m!$ sequences. Existing selection methods are based on parametric models. However, it is difficult to determine a good approximate model for an ordering problem before collecting the experimental data. With this in mind, we propose a method for choosing the subset for searching for the optimal order without assuming a prespecified model. The proposed method explores the inherent characteristics of the possible orders by using the distance between the positions of the components. We propose a systematic construction method for selecting a subset with a flexible run size, and also show its optimality. Compared with existing model-based methods, the proposed method is more appropriate when the model choice is not clear a priori.

Key words and phrases: Design of experiments, fractional order of addition design, pair-wise ordering distance.

1. Introduction

A wide variety of practical problems involve ordering, including scheduling problems and order of addition (OofA) experiments. The goal of such problems is to find the optimal sequence among all possible orderings. An important feature of an ordering problem is that the performance of the response may depend on the order of the m components. The scheduling problem is an optimization problem. The classical job scheduling problem sequences m jobs under given constraints, and its purpose is to find the optimal sequence that minimizes the total penalty. The scheduling problem can be viewed as an application of the ordering problem. For comprehensive discussions on scheduling problems, can refer to Townsend (1978), Leung (2004), Pinedo (2016), Hermelin et al. (2019), Wei (2019), and the references therein. More OofA experiments can be found in Fuleki and Francis (1968), Shinohara and Ogawa (1998), Karim, McCormick and Kappagoda (2000), Ding et al. (2015), Lin and Peng (2019),

Corresponding author: Yongdao Zhou, School of Statistics and Data Science, LPMC & KLMDASR, Nankai University, Tianjin 300071, China. E-mail: ydzhou@nankai.edu.cn.

Peng, Mukerjee and Lin (2019), Voelkel (2019), Chen, Mukerjee and Lin (2020), Mee (2020), Winker, Chen and Lin (2020), Yang, Sun and Xu (2021), Chen, Peng and Lin (2021), and Zhao, Lin and Liu (2021), among others. Other applications of ordering problems can be found in the bio-chemistry, nutritional science, pharmaceutical sciences, and engineering.

Usually, performing all $m!$ experiments is infeasible, because $m!$ is typically large. For example, $10! \approx 3.6$ million. Hence, we need to select a subset of the possible $m!$ orders to reduce the computational burden when determining the optimal order. Experimental designs are efficient procedures for planning experiments so that the obtained design can be analyzed to yield valid and objective conclusions, with substantially fewer experimental runs.

This study seeks a new direction for ordering problems by designing of experiments without assuming a prespecified model. Denote the $m! \times m$ matrix of all such permutations as F_m. Here, F_m is the full design, and a subset of its rows is a fractional design. We propose a novel approach for selecting a subset of F_m, such that the subset keeps as much of the information of F_m as possible. For selecting a subset of F_m, existing selection methods are based on approximate models, such as the pair-wise ordering (PWO) model (van Nostrand (1995); Voelkel (2019)) and component position (CP) model (Yang, Sun and Xu (2021); Stokes and Xu (2022)). However, there is no evidence that the approximate model reveals the true relationship between the effect of the process order and the response if one has no prior information. This motivates our development of a novel method of exploring the inherent characteristics of all $m!$ possible orders F_m, because it is necessary to consider a model-free method in order to select the most informative design.

The main idea of our construction method for such a fractional design is to keep as much information of the full design F_m as possible. To satisfy this requirement, we first study the inherent characteristics of F_m. Next, we determine the exact distances between all components in each row, and then obtain a pair-wise ordering distance (PWOD) array from all possible ordering designs. Some new criteria are proposed to measure the similarity between a fractional design and the full design F_m. Based on the criteria, we propose a construction method for model-free fractional designs with a flexible run size n.

The rest of the paper is organized as follows. Section 2 introduces the terminology. Some optimality criteria are considered in Section 3. A construction method for model-free fractional OofA designs is proposed in Section 4. The theoretical properties of the obtained fractional designs are shown in Section 5. Section 6 presents two case studies that illustrate the usefulness of the proposed

Table 1. The full design F_4 and the PWOD array P_4.

	F_4	P_4^*			$P_4 \backslash P_4^*$				F_4	P_4^*			$P_4 \backslash P_4^*$		
Run	Order	δ_{01}	δ_{02}	δ_{03}	δ_{12}	δ_{13}	δ_{23}	Run	Order	δ_{01}	δ_{02}	δ_{03}	δ_{12}	δ_{13}	δ_{23}
1	**0123**	**1**	**2**	**3**	1	2	1	13	**0312**	**2**	**3**	**1**	1	−1	−2
2	1023	−1	1	2	2	3	1	14	1302	−2	1	−1	3	1	−2
3	0213	2	1	3	−1	1	2	15	0321	3	2	1	−1	−2	−1
4	**1203**	**−2**	**−1**	**1**	1	3	2	16	**1320**	−3	**−1**	**−2**	2	1	−1
5	**2013**	**1**	**−1**	**2**	−2	1	3	17	**2301**	**1**	**−2**	**−1**	−3	−2	1
6	2103	−1	−2	1	−1	2	3	18	2310	−1	−3	−2	−2	−1	1
7	0132	1	3	2	2	1	−1	19	3012	1	2	−1	1	−2	−3
8	**1032**	**−1**	**2**	**1**	3	2	−1	20	**3102**	**−1**	**1**	**−2**	2	−1	−3
9	**0231**	**3**	**1**	**2**	−2	−1	1	21	**3021**	**2**	**1**	**−1**	−1	−3	−2
10	1230	−3	−2	−1	1	2	1	22	3120	−2	−1	−3	1	−1	−2
11	2031	2	−1	1	−3	−1	2	23	3201	1	−1	−2	−2	−3	−1
12	**2130**	**−2**	**−3**	**−1**	−1	1	2	24	**3210**	**−1**	**−2**	**−3**	−1	−2	−1

Note: P_4^* denotes the first three columns in P_4; the bold symbols are explained in Section 4.

method. Concluding remarks are given in Section 7. The proofs of the propositions and theorems are provided in the Supplementary Material.

2. PWOD Array

Denote $\Phi(i)$ as the position of component i in a permutation of m components, $Z_m = \{0, 1, \ldots, m-1\}$. Let $\delta_{ij} = \Phi(j) - \Phi(i)$ be the directed distance between the components i and j. Thus, each permutation on Z_m is one to one, and determined by the $m-1$ directed distances δ_{ij}, for a given i. For example, for the order $\{1, 0, 2, 3\}$, we have $\delta_{01} = -1, \delta_{02} = 1$, and $\delta_{03} = 2$. Considering all pairs (i, j) with $1 \le i < j \le m$, we obtain a new PWOD array. For simplicity, we denote the PWOD array with m components as P_m. Note that P_m is an $m! \times (m(m-1)/2)$ matrix and the pairs are ordered lexicographically. We want to select n runs from among all possible $m!$ orders (runs), and keep the balance of the levels of each column in the PWOD array. The optimal value of n is a trade-off between the computational cost and information loss.

The directed distance $\delta_{ij} \in \{\pm 1, \pm 2, \ldots, \pm(m-1)\}$ indicates whether component i is added before component j, as well as the distance between the two components. For any design D, we can obtain the corresponding PWOD array, and vice versa. Table 1 shows the case of $m = 4$.

Lemma 1. *The PWOD array P_m is one to one, and determined by the full design matrix F_m.*

Next, we show the properties of the PWOD array of the full design F_m, which can be used to construct optimal fractional designs with flexible run sizes. The PWOD array in Table 1 shows that, in every column each of the terms "3" and "−3" appears twice, each of the terms "2" and "−2" appears four times, and each of the terms "1" and "−1" appears six times. Denote $E_k(a)$ as the frequency of the term a in the kth column of the PWOD array P_m. Then, we have the following result.

Proposition 1. *In each column of the PWOD array P_m, the term a occurs $E(a) = (m - |a|) \times (m-2)!$ times, for all $a \in \{\pm 1, \ldots, \pm(m-1)\}$.*

The result in Proposition 1 is used in the criterion χ^2_P defined in Section 3. Moreover, for the PWOD array P_m, we have the following result.

Proposition 2. *The rank of P_m is $m-1$.*

Proposition 2 shows that the position of each element is determined by selecting the $m-1$ linearly independent columns of P_m that contain all m elements, and the designs are invariant for the choice of the $m-1$ columns. Without loss of generality, we select the first $m-1$ columns that contain the component zero, denoted by P^*_m. For example, the PWOD array P^*_4 is formed by the first three columns δ_{01}, δ_{02}, and δ_{03} in Table 1. Here, we need only to consider the PWOD array P^*_4, instead of the full array P_m. Moreover, for an n-run fractional design from F_m, denote the first $m-1$ columns of the corresponding PWOD array by $P^*_{m,n}$. A covering array $CA(N, k, v, t)$ is an $N \times k$ array with entries from a set X of v symbols, such that every $N \times t$ sub-array contains all t-tuples over X at least once, where t is the strength of the array, and v is the number of symbols for each column (Yin (2003)). From the definition of a covering array, the PWOD array P^*_m can be considered as a covering array $CA(m!, m-1, 2(m-1), 1)$.

Next, we explore the frequencies of pairs of values appearing in any two columns of the PWOD array P^*_m. The following result gives the frequencies of two columns in the PWOD P^*_m.

Proposition 3. *In any two-column sub-array of the PWOD array P^*_m, we have (i) for any $a \in \{1, \ldots, m-2\}$, the pair $(m-a, b)$ occurs $a(m-3)!$ times when $b = 1, \ldots, m-a-1$, and the pair $(-(m-a), b)$ occurs $a(m-3)!$ times when $b = -(m-a-1), \ldots, -1$; (ii) for any $g \in \{2, \ldots, m-1\}$, for $j = 1, \ldots, g-1$, the pair $(m-g, b)$ occurs $(g-j)(m-3)!$ times when $b = -j$ or $(m-g+j)$, and the pair $(-(m-g), b)$ occurs $(g-j)(m-3)!$ times when $b = j$ or $-(m-g+j)$.*

Proposition 3 shows the combinational property for any two columns in the PWOD array P^*_m. For any t columns of the PWOD array P^*_m, each t-tuple is

determined by the $(t+1)$ components, and each combination appears a multiple of $(m-(t+1))!$ times, $t \in \{1, \ldots, m-1\}$.

Corollary 1. *For any t-tuple of the sub-array in the PWOD array P_m^*, the greatest common divisor of the number of all terms in P_m^* is $(m-(t+1))!$, $t \in \{1, \ldots, m-1\}$.*

Corollary 1 shows that the run size of a subset that includes all terms in P_m^* is $(m-(t+1))!$. Note that it is infeasible to perform all $m!$ permutations when m is large. Thus, a subset of the $m!$ possible orders should be selected to find the optimal order. To select n rows from the full OofA experiment F_m, we have the following requirements:

(i) The $n \times m$ fractional design must be a balanced design;

(ii) Each column of the $n \times (m-1)$ PWOD array $P_{m,n}^*$ contains all possible terms that appear in P_m^*, and the number of each term should be proportional to that in P_m^*.

To satisfy the two requirements, the run size n of the fractional design can be chosen as the ratio between $m!$ and the greatest common divisor of the number of all terms in P_m^*. Moreover, for any $t \in \{1, \ldots, m-1\}$, if each $n \times t$ sub-array of $P_{m,n}^*$ contains all possible t-tuple combinations in P_m^*, the smallest run size n is

$$n = \frac{m!}{gcd(terms)} = \frac{m!}{(m-(t+1))!} = m(m-1)\cdots(m-t),$$

where "gcd" means the greatest common divisor. Thus, we have the following result for choosing the smallest n that achieves the requirements (i) and (ii).

Theorem 1. *For any $t \in \{1, \ldots, m-1\}$, the smallest run size n is equal to $m(m-1)\cdots(m-t)$, such that each $n \times t$ sub-array of $P_{m,n}^*$ contains all possible t-tuple combinations in P_m^*. In particular, we have $n = m(m-1)$ when $t = 1$, and $n = m!$ when $t = m-1$.*

Theorem 1 shows that the run size n increases exponentially with t. Thus, the choice of $t = 1$ is preferable. Moreover, if every column of $P_{m,n}^*$ contains all possible terms in P_m^*, the run size of the selected design must be a multiple of $m(m-1)$. It is evident that the component orthogonal arrays proposed by Yang, Sun and Xu (2021) have a multiple of $m(m-1)$ runs. In particular, $m(m-1)$ is the smallest run size n satisfying requirements (i) and (ii) with strength $t = 1$. To select a more flexible run size, one may consider n that is a multiple of $m-1$, although this may sacrifice requirement (ii).

3. Optimality Criteria

For a given linear model $y = X\beta + \varepsilon$, the D-efficiency is a proper criterion that reflects the capacity of the estimation. Define the D-value of a design D by $\mathrm{D}_e(D) = (1/n)|X^TX|^{1/q}$, where X is a model matrix with respect to the prespecified model, q is the number of columns of X, and n is the run size of X. Define the D-efficiency of a fractional design D as the ratio of its D-value over the D-value of the full design, that is, $RD(D) = \mathrm{D}_e(D)/\mathrm{D}_e(F_m)$.

Despite the popularity of model-based criteria (e.g., D-efficiency, A-efficiency, etc.), their performance is likely to depend on the difference between the true model and the assumed model. For the ordering problem, it is challenging to find a good approximate model that reflects the true relationship between the order effects and the responses. To increase the probability of identifying an optimal order that will be robust to a large variety of models, we should use an alternate criterion to collect design points that can be used to build as many models as possible, unless one has prior information about the model.

Some criteria exist for evaluating the optimality of fractional designs, such as the $\chi^2(D)$ criterion defined by Yamada and Lin (1999), $E(f_{NOD})$ criterion proposed by Fang, Lin and Liu (2003), and $E(\chi^2)$ criterion introduced by Ai, Fang and He (2007). Let $D = (d_{ij})$ be an $n \times m$ matrix of a mixed-level design. The $\chi^2(D)$ criterion minimizes $\chi^2_{ave}(D) = \sum_{1\le i<j\le m} \binom{m}{2}^{-1} \chi^2_{i,j}(D)$, where $\chi^2_{i,j}(D) = \sum_{a=1}^{s_i}\sum_{b=1}^{s_j}[n_{i,j}(a,b) - n/(s_is_j)]^2/(n/(s_is_j))$, s_i is the number of levels in the ith column of the design D, and $n_{i,j}(a,b)$ is the number of (a,b) pairs in the pair of columns (x^i, x^j). It is known that $\chi^2_{i,j}(D)$ measures the balance relative to the pair of columns (x^i, x^j), and $\chi^2_{ave}(D)$ is an overall measure of nonorthogonality.

Motivated by the idea of the $\chi^2(D)$ criterion, we generalize $\chi^2_{i,j}(D)$ to measure the balance relative to any t columns out of m on the original OofA design. For an n-run OofA design D, define

$$\chi^2_F(D) = \sum_{1\le i<j\le m} \binom{m}{2}^{-1} \chi^2_{i,j}(D), \quad \text{and}$$

$$\chi^2_{i,j}(D) = \sum_{\substack{a\neq b \\ a,b\in Z_m}} \frac{[n_{i,j}(a,b) - n/(m(m-1))]^2}{n/(m(m-1))}. \tag{3.1}$$

For an OofA design D, $n_{i,j}(a,b) - n/(m(m-1))$ represents the difference between the actual value and the average value of $n_{i,j}(a,b)$. For any i, j, a, and b, $n_{i,j}(a,b) = (m-2)!$ and $n = m!$ for the full design F_m, which implies that $\chi^2_F(F_m) = 0$. A smaller $\chi^2_F(D)$ value implies a more balanced design. In partic-

ular, the design D is a balanced design if $\chi_F^2(D) = 0$.

Furthermore, we extend $\chi_{i,j}^2(D)$ to measure the balance relative to any t columns in the model matrix with any model. For simplicity, we show only the $\chi_P^2(D)$ based on the PWOD array. The criteria on other arrays (in the literature) can be obtained similarly. Define

$$\chi_P^2(D) = \frac{1}{m-1} \sum_{k=1}^{m-1} \sum_{a \in \{\pm 1, \ldots, \pm(m-1)\}} \frac{[n_k(a) - nE_k(a)/m!]^2}{nE_k(a)/m!}, \tag{3.2}$$

where $n_k(a)$ and $E_k(a)$ are the frequencies of the term a in the kth column of the PWOD arrays of $P_{m,n}$ and P_m, respectively. The $\chi_{\mathrm{P}}^2(D)$ criterion is invariant to the choice of the $m-1$ columns of the PWOD array. Without loss of generality, the $\chi_P^2(D)$ criterion considers only the first $m-1$ columns in (3.2). Then, $\chi_F^2(D)$ and $\chi_P^2(D)$ reflect the balance between any two columns and the $m-1$ columns of the fractional design D, respectively. Moreover, the two criteria do not depend on a prespecified model; that is, they are model free. Obviously, the smaller the value of $\chi_F^2(D)$ or $\chi_P^2(D)$, the better the fractional design D keeps the same balance as that of the full design F_m. The proposed criteria $\chi_F^2(D)$ and $\chi_P^2(D)$ distinguish between the distances from one component to another, and reward designs in which these distances are balanced.

4. Construction Method of Fractional OofA Design

In this section, we propose a construction method for choosing a subset from the full design such that its run size is a multiple of $m-1$, where m is a prime or prime power. All operations in the construction method are built on the Galois field. The construction method is based on the difference matrix, which has many desired properties (see van Greevenbroek and Jedwab (2018)). Let $GF(v)$ be a Galois field of order v. The difference matrix $DM(n,k,v)$ is an $n \times k$ array $A = (a_{ij})$ with entries from $GF(v)$, such that for the lth and hth columns of A, $(1 \le l < h \le k)$, the difference list $\{a_{ih} - a_{il}\}$ contains every element of $GF(v)$ the same number of times. Additional information on the difference matrix can be found in Yin (2003). The designs obtained by the proposed method do not depend on a prespecified model. The proposed method can be employed when m is a prime number $(m > 2)$ or a prime power.

For a prime power $m = s^p$, s is a prime number and p is a positive integer, let $GF(s) = \{0, 1, \ldots, s-1\}$, and $GF(m) = \{\mu_0 = 0, \mu_1 = 1, \mu_2, \ldots, \mu_{m-1}\}$ denote Galois fields. Denote $\mathbf{x}_i = (0, 1, x_{i,3}, \ldots, x_{i,m})$ and $(x_{i,3}, x_{i,4}, \ldots, x_{i,m})$ as permutations of $\{\mu_2, \ldots, \mu_{m-1}\}$. For an $n \times m$ matrix $A = (a_{ij})$ and a column

vector $\mathbf{b} = (b_1, \ldots, b_n)^T$, define $A \oplus_c \mathbf{b} = (a_{ij} + b_i)$ on $GF(m)$. Let $\mathbf{1}_{m-1}$ and $\mathbf{0}_{m-1}$ be the $(m-1) \times 1$ vectors with all elements one and zero, respectively. Denote $\lfloor c \rfloor$ as the greatest integer not exceeding c. The main idea of the construction method is as follows. First, a difference matrix $DM_{i,1} = (\mathbf{x}_i^T, \mu_2 \mathbf{x}_i^T, \ldots, \mu_{m-1}\mathbf{x}_i^T)^T$ can be generated from a given $\mathbf{x}_i$, and several difference matrices can also be obtained based on $DM_{i,1}$. Next, we obtain other $1 \times m$ row vectors by permuting the last $m-2$ elements of $\mathbf{x}_i$, as well as other difference matrices. Combining the difference matrices, we obtain various $n \times m$ designs. The best design is obtained under some optimality criteria. The detailed procedure is shown in Algorithm 1.

Algorithm 1 Construction method of fractional designs when m is a prime power.

Step 1: Given: the optimal criterion $\phi(\cdot)$ and $n, m = s^p$, where s is a prime number and p is a positive integer. Let $k_1 = \lfloor n/(m(m-1)) \rfloor$ and $k_2 = (n/(m-1)) - k_1 m$.

Step 2: For $i = 1, 2, \ldots, k_1 + 1$, let $\{\pi_3, \ldots, \pi_m\}$ be one of the $(m-2)!$ permutations of $\{\mu_2, \ldots, \mu_{m-1}\}$, $\mathbf{x}_i = (0, 1, \pi_3, \ldots, \pi_m)$, and $DM_{i,1} = (\mathbf{x}_i^T, \mu_2 \mathbf{x}_i^T, \ldots, \mu_{m-1}\mathbf{x}_i^T)^T$. Then, $DM_{i,k} = [DM_{i,1} \oplus_c (s-1) DM_{i,1}^k]$, where $DM_{i,1}^k$ is the kth column of $DM_{i,1}$, for $k = 2, \ldots, m$. Let $DM_i = ({DM_{i,1}}^T, \ldots, {DM_{i,m}}^T)^T$.

Step 3: If $k_2 = 0$, obtain the $n \times m$ design matrix $D = ({DM_1}^T, {DM_2}^T, \ldots, {DM_{k_1}}^T)^T$. If $k_2 \neq 0$, denote $DM_{k_1+1} = ({DM_{k_1+1,j_1}}^T, \ldots, {DM_{k_1+1,j_{k_2}}}^T)^T$, where $j_1, \ldots, j_{k_2} \in \{1, \ldots, m\}$. Then, obtain the $n \times m$ design matrix $D = ({DM_1}^T, \ldots, {DM_{k_1}}^T, {DM_{k_1+1}}^T)^T$.

Step 4: Search for the optimal design D^* among all possible D under the criterion $\phi(\cdot)$.

In Step 1, the optimal criterion can be chosen as the $\chi_P^2(D)$ or the D-efficiency with a given model. In Step 2, the total number of possible x_i is $(m-2)!$. Each DM_i has $m(m-1)$ rows. If $k_2 = 0$, we choose k_1 different $\mathbf{x}_i$ to form D, and have $N = \binom{(m-2)!}{k_1}$ possible designs in Step 3. If $k_2 \neq 0$, we select $\mathbf{x}_1, \ldots, \mathbf{x}_{k_1+1}$ from the $(m-2)!$ possible $\mathbf{x}_i$ to obtain $DM_1, \ldots, DM_{k_1}, DM_{k_1+1}$, and then choose k_2 blocks from the m blocks in DM_{k_1+1}. Therefore, there are $N = (k_1 + 1)\binom{(m-2)!}{k_1+1}\binom{m}{k_2}$ possible choices. Step 4 compares the N designs under the optimal criterion $\phi(D)$ to select the best one. On the other hand, if we directly choose n runs from the $m!$ runs of the full design, we need to compare $\binom{m!}{n}$ designs to determine the best one. Obviously, N is much smaller than $\binom{m!}{n}$, even for small m. For example, when $m = 4$ and $n = 9$, $\binom{m!}{n} = 1307504$, whereas $N = (0+1)\binom{(4-2)!}{0+1}\binom{4}{3} = 8$. It is clear that Algorithm 1 significantly decreases the computational complexity.

The following example illustrates the construction procedure for the case of

$m = 4$.

Example 1. Let $m = 4$, $GF(2) = \{0, 1\}$, and $GF(4) = \{0, 1, x, x+1\}$. Consider the following four cases: (i) $n = 6$, and thus $k_1 = 0$ and $k_2 = 2$; (ii) $n = 9$, and thus $k_1 = 0$ and $k_2 = 3$; (iii) $n = 12$, and thus $k_1 = 1$ and $k_2 = 0$; (iv) $n = 15$, and thus $k_1 = 1$ and $k_2 = 1$. Compared with all possible d_i, when $\mathbf{x}_1 = (0, 1, x, x+1)$, the difference matrix

$$DM_{1,1} = \begin{pmatrix} 0 & 1 & x & x+1 \\ 0 & x & x+1 & 1 \\ 0 & x+1 & 1 & x \end{pmatrix}$$

is the best when comparing all possible $(m-2)!$ difference matrices. From Step 2 of Algorithm 1, we have

$$DM_{1,2} = [DM_{1,1} \oplus_c (2-1) DM_{1,1}^2] = DM_{1,1} \oplus_c \begin{pmatrix} 1 \\ x \\ x+1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x+1 & x \\ x & 0 & 1 & x+1 \\ x+1 & 0 & x & 1 \end{pmatrix},$$

in the sense of modulo 2. Similarly, $DM_{1,3}$ and $DM_{1,4}$ can be obtained as follows:

$$DM_{1,3} = \begin{pmatrix} x & x+1 & 0 & 1 \\ x+1 & 1 & 0 & x \\ 1 & x & 0 & x+1 \end{pmatrix}, \text{ and } DM_{1,4} = \begin{pmatrix} x+1 & x & 1 & 0 \\ 1 & x+1 & x & 0 \\ x & 1 & x+1 & 0 \end{pmatrix}.$$

When replacing $\{0, 1, x, x+1\}$ with $\{0, 1, 2, 3\}$ for each case, the desired designs result. For instance, when $n = 6$ and $k_2 = 2$, we should select two difference matrices and find that $D = ({DM_{1,1}}^T, {DM_{1,3}}^T)^T$ is a six-run optimal design with a minimum $\chi_P^2(D)$ of 1.333. It can be verified that $D = ({DM_{1,1}}^T, {DM_{1,2}}^T, {DM_{1,4}}^T)^T$ is a nine-run optimal design with a minimum $\chi_P^2(D)$ of 0.556. Let $D = ({DM_{1,1}}^T, {DM_{1,2}}^T, {DM_{1,3}}^T, {DM_{1,4}}^T)^T$ be a 12×4 matrix. By calculating the $\chi_P^2(D)$ of all possible designs with 12 runs, we find that D is the optimal design with a minimum $\chi_P^2(D)$ of zero. From Lemma 1, for $t = 1$, the PWOD array is a covering array $CA(12, 3, 6, 1)$, shown in Table 1 in bold. When $n = 15$, we should select a design with $m(m-1) = 12$ runs and pick another ($k_2 = 1$) difference matrix. By Step 2 in Algorithm 1, we have $D_1 = ({DM_{1,1}}^T, {DM_{1,2}}^T, {DM_{1,3}}^T, {DM_{1,4}}^T)^T$. Evaluating the remaining possible $\mathbf{x}_i$, we choose $\mathbf{x}_2 = (0, 1, x+1, x)$, yielding $DM_{2,1} = (d_2^T, \mu_2 d_2^T, \ldots, \mu_{m-1} d_2^T)^T$, and $DM_{2,2} = [DM_{2,1} \oplus_c (2-1) DM_{2,1}^2] = [DM_{2,1} \oplus_c (1, x, x+1)^T] = [(1, x, x+1)^T, (0, 0, 0)^T, (x, x+1, 1)^T, (x+1, 1, x)^T]$. Note that all elements in the second column of $DM_{2,2}$ are zero. It can be verified that the design $D = (D_1^T, DM_{2,2}^T)^T$

Table 2.

$\mathbf{DM}_{1,1}$					$\mathbf{DM}_{1,2}$					$\mathbf{DM}_{1,3}$					$\mathbf{DM}_{1,4}$					$\mathbf{DM}_{1,5}$					$\mathbf{DM}_{2,1}$					$\mathbf{DM}_{2,3}$				
0	1	4	2	3	4	0	3	1	2	1	2	0	3	4	3	4	2	0	1	2	3	1	4	0	0	1	2	3	4	3	4	0	1	2
0	2	3	4	1	3	0	1	2	4	2	4	0	1	3	1	3	4	0	2	4	1	2	3	0	0	2	4	1	3	1	3	0	2	4
0	3	2	1	4	2	0	4	3	1	3	1	0	4	2	4	2	1	0	3	1	4	3	2	0	0	3	1	4	2	4	2	0	3	1
0	4	1	3	2	1	0	2	4	3	4	3	0	2	1	2	1	3	0	4	3	2	4	1	0	0	4	3	2	1	2	1	0	4	3

is an optimal one with a minimum $\chi^2_P(D)$ of 0.333. Different criteria may yield different corresponding optimal designs.

Algorithm 1 can also be used to construct the OofA design with prime m. The following example illustrates the construction procedure for the case of $m = 5$.

Example 2. For $m = 5$, let $\phi(D)$ be $\chi^2_P(D)$. Consider the following four cases: (i) $n = 12$, and thus $k_1 = 0$ and $k_2 = 3$; (ii) $n = 16$, and thus $k_1 = 0$ and $k_2 = 4$; (iii) $n = 20$, and thus $k_1 = 1$ and $k_2 = 0$; (iv) $n = 24$, and thus $k_1 = 1$ and $k_2 = 1$. By using the exhaustive method in Step 2, the best difference matrix is $DM_{1,1} = (\mathbf{x}_1^T, 2\mathbf{x}_1^T, \ldots, (m-1)\mathbf{x}_1^T)^T$, the first run is $\mathbf{x}_1 = (0, 1, 4, 2, 3)$. Furthermore, from the remaining possible $\mathbf{x}_i$, $\mathbf{x}_2 = (0, 1, 2, 3, 4)$ is chosen, resulting in $DM_{2,1} = (\mathbf{x}_2^T, 2\mathbf{x}_2^T, \ldots, (m-1)\mathbf{x}_2^T)^T$ and the corresponding DM_2. The difference matrices $DM_{1,1}$, $DM_{1,2}$, $DM_{1,3}$, $DM_{1,4}$, $DM_{1,5}$, $DM_{2,1}$, and $DM_{2,3}$ are shown below.

Next, we obtain the desired designs for each case, as follows. (i) When $n = 12$, $D_1 = ({DM_{1,1}}^T, {DM_{1,3}}^T, {DM_{1,5}}^T)^T$ is found to be the optimal design, with a $\chi^2_P(D)$ of 0.778, which is the minimum value among all of the 12-run designs based on the exhaustive method. (ii) When $n = 16$, from Steps $4 - 5$, $D_2 = ({DM_{1,1}}^T, {DM_{1,2}}^T, {DM_{1,4}}^T, {DM_{1,5}}^T)^T$ is found to be the optimal design with $4(m-1)$ runs, which achieves the minimum $\chi^2_P(D)$ of 0.458. (iii) When $n = 20$, let $D_3 = DM_1 = ({DM_{1,1}}^T, \ldots, {DM_{1,5}}^T)^T$. Calculating the $\chi^2_P(D)$ values of all possible $n_1 = \binom{3!}{1}$ designs, we find that D_3 is the optimal design, with a minimum $\chi^2_P(D)$ of zero. From Lemma 1, the corresponding PWOD array is the covering array $CA(20, 4, 8, 1)$. (iv) When $n = 24$, the design $D_4 = (D_3^T, {DM_{2,3}}^T)^T$ is optimal, achieving a minimum $\chi^2_P(D)$ of 0.306, where $DM_{2,3}$ makes the third column of $DM_{2,1}$ become $\mathbf{0}_{m-1}$.

Note that the construction method for the difference matrix in Algorithm 1 is not suitable for the case $m = 6$. By comparing all possible $(6-2)! = 24$ matrices, we provide the following three possible 5×6 initial design matrices:

$$\begin{pmatrix} 0\,1\,2\,5\,3\,4 \\ 0\,2\,1\,4\,5\,3 \\ 0\,3\,5\,1\,2\,4 \\ 0\,4\,3\,1\,5\,2 \\ 0\,5\,1\,4\,3\,2 \end{pmatrix}, \quad \begin{pmatrix} 0\,1\,4\,5\,3\,2 \\ 0\,1\,5\,2\,4\,3 \\ 0\,3\,2\,5\,1\,4 \\ 0\,4\,1\,3\,5\,2 \\ 0\,5\,3\,4\,2\,1 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 0\,1\,2\,5\,3\,4 \\ 0\,2\,3\,5\,4\,1 \\ 0\,3\,4\,2\,1\,5 \\ 0\,4\,2\,5\,1\,3 \\ 0\,5\,1\,4\,3\,2 \end{pmatrix}. \tag{4.1}$$

Using the above initial matrices as $DM_{i,1}$ in Step 2 of Algorithm 1, we obtain the corresponding fractional OofA designs.

5. Properties of the Constructed Designs

In this section, we discuss the properties of the designs constructed using Algorithm 1. Because the full design F_m is optimal (Peng, Mukerjee and Lin (2019)), we are particularly interested in the "similarity" between the constructed design and the full design, which can be measured using the criteria $\chi^2_P(D)$ and $\chi^2_F(D)$. From Algorithm 1, we have the following result.

Theorem 2. *When $n = \prod_{i=0}^{t}(m-i)$, for $t \in \{1, 2, \ldots, m-1\}$, the fractional design constructed using Algorithm 1 is a balanced design. In particular, when $t = 1$, the corresponding PWOD array of the $m(m-1)$-run fractional design D constructed using Algorithm* 1 *is a covering array $CA(m(m-1), m-1, 2(m-1), 1)$; that is, every column of the PWOD array P^*_m contains all terms $\{\pm 1, \ldots, \pm(m-1)\}$ at least once.*

We next discuss the "similarity" between the full design F_m and the designs constructed using Algorithm 1 in the sense of distance. For a design D, denote the Hamming distance $d_H(a, b)$ of any two rows $\mathbf{a}$ and $\mathbf{b}$ as the number of places by which they differ. Define $d_H(D) = \min\{d_H(\mathbf{a}, \mathbf{b}), \mathbf{a} \in D, \mathbf{b} \in D, \mathbf{a} \neq \mathbf{b}\}$ as the minimum Hamming distance between all design points. Then, we have the following result.

Theorem 3. *Any n-run design D constructed using Algorithm* 1*, for $2 \leq d_H(D) \leq m-1$, achieves the upper bound of the Hamming distance when $n = k(m-1)$, for $k = 1, \ldots, m$.*

Theorem 3 shows that the proposed method constructs the maximum Hamming distance design with a run size of $n = k(m-1)$, for $k = 1, \ldots, m$. Fang, Ge and Liu (2002) showed that the Hamming distance is closely related to the discrete discrepancy, which is an important uniformity criterion in the theory of uniform design. Thus our designs have a good space-filling property. Wiens (1991) showed that uniform designs are robust when the model is unknown.

Therefore, uniformity is a good criterion for choosing a model-free fractional design (uniform design), which can be used in a preliminary exploratory analysis when the exact model is unknown.

Next, we consider the constructed designs under the criteria $\chi^2_P(D)$ and $\chi^2_F(D)$. For the $\chi^2_P(D)$ criterion, we have the following result.

Theorem 4. *When the run size $n = km(m-1)$, for $k = 1, 2, \ldots, (m-2)!$, the constructed designs have $\chi^2_P = 0$.*

Theorem 4 shows that the obtained design is optimal under the $\chi^2_P(D)$ criterion when n is a multiple of $m(m-1)$. If n is not a multiple of $m(m-1)$, the χ^2_P value does not equal zero and can be used to identify the property of different designs. For the $\chi^2_F(D)$ criterion, we have the following result.

Theorem 5. *Given n and m, the design constructed using Algorithm 1 achieves the minimum $\chi^2_F(D)$ value, $m(m-1)\left[c - n^2/(m(m-1))\right]/n$, where $c = (m-1)[(k_1+1)^2k_2+k_1^2(m-k_2)]$ and k_1 and k_2 are defined in Algorithm 1. Furthermore, $\chi^2_F(D) = 0$ for any $n = km(m-1)$, for $k = 1, 2, \ldots, (m-2)!$.*

Theorem 5 shows that the obtained designs are similar to the full design in terms of the $\chi^2_F(D)$ criterion. From the proof of Theorem 5 in the Supplementary Material, we easily obtain the following result.

Corollary 2. *Given n and m, all designs constructed using Algorithm 1 have the same $\chi^2_F(D)$ value, and all achieve the minimum value.*

From Theorem 5 and Corollary 2, all the constructed designs are optimal under the criterion $\chi^2_F(D)$. In other words, the designs constructed using the proposed method have a good property of balance. The case studies in Section 6 show that the properties based on the proposed criteria are deemed acceptable for efficiently creating good OofA experiments.

6. Case Studies

In this section, we present case studies based on the OofA experiment and the job scheduling problem to illustrate the usefulness of the proposed designs.

OofA experiments are popular when the response of interest is affected by the sequence of materials or components are added. The objective of an OofA experiment is to find the optimal addition order. Here, we present a case study with $m = 4$ to show the procedure of finding the optimal order. We use data from the drug combination experiment in Yang, Sun and Xu (2021) to illustrate

the usefulness of the PWOD arrays. For OofA experiments, the following models can be used as the approximate model.

For any pair of components i and j, van Nostrand (1995) proposed an order effect, a "pseudo factor", known as the PWO factor. Let $I_{ij} = 1$ if i precedes j, and -1 otherwise. The first approximate model is called the PWO model:

$$y = \beta_0 + \sum_{i<j} \beta_{ij} I_{ij} + \varepsilon, \tag{6.1}$$

where y is the response of interest, I_{ij} is the PWO factor, and ε is a random error.

Another model is the CP model proposed by Yang, Sun and Xu (2021), used to determine which component should be added in each specific position. The CP model is

$$y_i = \mu_0 + \sum_{c=0}^{m-1} \sum_{j=1}^{m} x_c^{(j)} \tau_c^{(j)} + \varepsilon_i, \tag{6.2}$$

where y_i is the response at the ith run, μ_0 is the overall mean, $x_c^{(j)} = 1$ if the component c is placed at position j, and zero otherwise, $\tau_c^{(j)}$ is the effect of component c at the jth position, and $\epsilon_i \sim N(0, \sigma^2)$ is an independent error. To make the model estimable, additional constraints are required, namely, $\tau_1^{(j)} = 0$ for $j = 1, \ldots, m$, and $\tau_c^{(m)} = 0$ for $c = 0, \ldots, m-1$.

Peng, Mukerjee and Lin (2019) considered a tapering PWO model that incorporates the order of each pair of components, and can also account for the distance between the two components in every such pair. The tapering PWO model is

$$y = \beta_0 + \sum_{i<j} \beta_{ij} z_{ij} + \varepsilon, \tag{6.3}$$

where $z_{ij} = c_{h(ij,a)}$ if i precedes j, and $-c_{h(ij,a)}$ otherwise, and $h(ij, a)$ is the distance between i and j in a; that is, if $a_k = i$ and $a_l = j$, then $h(ij, a) = |k - l|$, and thus $h(ij, a) \in \{1, \ldots, m-1\}$. Common choices of c_h are $c_h = 1/h$ or $c_h = c^{h-1}$ with known c, for $0 < c < 1$, and $h \in \{1, \ldots, m-1\}$. Here, we choose $c_h = 1/h$. Mee (2020) considered higher-order models for OofA experiments. However, a high-order model includes too many parameters to be estimated.

Apart from the above models, based on the PWOD array, we also consider the first-order PWOD linear model

$$y = \beta_0 + \sum_{j=1}^{m-1} \beta_j \delta_{0j} + \epsilon, \tag{6.4}$$

where y is the response of interest, δ_{0j} is the jth column of the PWOD array $P^*_{m,n}$, and the random error $\epsilon \sim N(0, \sigma^2)$. Similarly, we also consider the approximate polynomial model including all linear, quadratic, and interaction terms. However, the model including all of the interaction terms of the PWOD array is not estimable. Similarly to Lin and Peng (2019), we delete the last interaction when estimating this model. Thus, the resulting model

$$y = \beta_0 + \sum_{j=1}^{m-1} \beta_j \delta_{0j} + \sum_{j=1}^{m-1} \beta_{jj} \delta_{0j}^2 + \sum_{1 \le j < k \le m-1, j \ne m-2} \beta_{jk} \delta_{0j} \delta_{0k} + \epsilon \qquad (6.5)$$

becomes estimable. The model 6.5 is called a second-order PWOD model.

In this example, $m = 4$ and all the responses Y of the $4! = 24$ orders are shown in Table S1 in the Supplementary Material. We construct the fractional designs with $n = 6, 9, 12, 15, 18, 21$ using Algorithm 1, and estimate the parameters in Model (6.4). For each model, we evaluate the fitted responses of the n runs and the predicted responses $\hat{Y}_n$ of the other $24 - n$ runs. Table S1 shows those values of $\hat{Y}_n$ for $n = 6, 9, 12, 15, 18, 21, 24$ under the first-order PWOD model.

From Table S1, the best order predicted by the models may not be identical to the real data, owing to random errors. For example, the predicted optimal order of the largest value in $\hat{Y}_6$ is "2031", whereas "0231" is the order of the true largest response, which is predicted as the third largest order in $\hat{Y}_6$. From Table S1, the identified optimal orders are $\{0231, 2301, 2031\}$. For comparison, the predicted values of $\hat{Y}_n$ for the models in (6.1), (6.2), (6.3) and (6.5) are also provided in the Supplementary Material. We find similar results for other models. Hence, the orders $\{0231, 2301, 2031\}$ can be used as the optimal sequences in this example.

Next, we consider the job scheduling problem, which plays an important role in manufacturing, production systems, and information processing environments Pinedo (2016). Consider m jobs requiring processing in a certain machine environment. The schedule hopes to sequence these jobs under some given constraints. The purpose of the job scheduling problem is to find the optimal sequence that minimizes the total penalty. Let p_i $(i = 1, \ldots, m)$ represent the processing time of job i on a machine, and $C_k(\pi) = \sum_{i=1}^{k} p_i$ be the completion time of the operation of job k in any permutation or order π. Suppose the total cost of any order π is denoted by $W(\pi) = \sum_{k=1}^{m} \omega_k (C_k(\pi))^2$, where ω_k is a prespecified weight.

Here, we consider a case in which seven jobs are to be sequenced, with a quadratic penalty function of its completion time. Without loss of generality, the prespecified weights and processing times of these jobs are generated randomly

Table 3. The optimal orders for the scheduling of seven jobs.

Order	Cost
$4 \to 6 \to 7 \to 3 \to 1 \to 2 \to 5$	2,156.139
$4 \to 7 \to 6 \to 3 \to 1 \to 2 \to 5$	2,159.902
$4 \to 6 \to 3 \to 7 \to 1 \to 2 \to 5$	2,181.422
$4 \to 6 \to 7 \to 2 \to 3 \to 1 \to 5$	2,257.665
$4 \to 7 \to 6 \to 2 \to 3 \to 1 \to 5$	2,261.429
$6 \to 4 \to 7 \to 2 \to 3 \to 1 \to 5$	2,290.255
$4 \to 7 \to 6 \to 3 \to 1 \to 5 \to 2$	2,293.191
$4 \to 7 \to 2 \to 6 \to 3 \to 1 \to 5$	2,317.848

from a χ_1^2 distribution, where $p = (9.688, 2.504,\ 1.981, 2.912, 6.688, 2.774, 8.314)$ and $\omega = (1.658, 0.340,\ 0.641, 2.515, 0.060,\ 1.112, 2.547)$. In this case, there are $7! = 5040$ possible job orders, which yield different costs. Using the proposed method, an optimal design with 24 runs is found and displayed in Table S3 in the Supplementary Material. The corresponding total costs are shown in the last column of Table S3.

For simplicity, the PWO model is used here as an approximate model for analyzing the job scheduling problem. Comparing the predicted values of 7!, we find some optimal orders, displayed in Table 3, with the sequence $4 \to 6 \to 7 \to 3 \to 1 \to 2 \to 5$ being the best.

For confirmation purposes, we evaluate all possible $7! = 5040$ orders. After computing the costs of all 5,040 orders, the minimum cost is found to be $\mathcal{W} = 2156.139$, with the corresponding order $4 \to 6 \to 7 \to 3 \to 1 \to 2 \to 5$. This is the optimal order identified using our method in Table 3. Note that other models such as the tapering PWO model, can also be used here. The results are similar, and are thus omitted here.

7. Conclusion

The response of an ordering problem depends on the order of adding m different components. The goal of an ordering problem is to find the optimal order(s) for optimizing the response. For example, the classical job scheduling problem (see the case study in Section 6) finds the optimal sequence to optimize the total penalty. Here, we have considered an ordering problem in which all m components are added sequentially, obtaining optimal orders by using the proposed method. For simplicity, we display the optimal order or several optimal orders if they are not unique. We have proposed a method to construct an optimal design without a prespecifical model. Based on the corresponding subsample

design points, one can consider all possible existing models to find the optimal order(s). In addition, the obtained designs perform well in terms of D-efficiencies under most existing models, such as the case of $m = 4$ displayed in Table 4.

We use the directed distance between each pair of components to represent the arrangement of OofA experiments. The position information of the components can be identified more clearly than the PWO factors. Based on the directed distance, the proposed PWOD arrays that account for the positions of each pair of components are more precise than the PWO arrays. Moreover, by the properties of PWOD arrays, the obtained designs contain all position information of the full design. Hence, the fractional OofA designs keep as much information about the full design as possible.

We provide an algorithm for constructing the fractional designs, the run sizes of which are multiples of $m - 1$. The proposed method can be extended to generate efficient designs of any run size (n) when one selects the first n rows of the design D constructed using Algorithm 1. This is more flexible than other construction methods in the literature. For example, the optimal designs constructed by Peng, Mukerjee and Lin (2019) often have larger run sizes, and the component orthogonal arrays proposed by Yang, Sun and Xu (2021) have at least $m(m - 1)$ runs. As previously mentioned, the proposed algorithm can be used for any prime number and prime power. For other cases, the corresponding algorithm should be studied further. From Theorem 1, the balanced design must have the smallest run size $n = \prod_{i=0}^{t}(m - i)$, for any $t \in \{1, \ldots, m - 1\}$. The proposed method results in a balanced design if the run size n is a multiple of $m(m - 1)$; otherwise, the proposed method is not balanced. Note that the goal of an ordering problem is to find the optimal order. In this study, we propose a method for choosing the subset for searching for the optimal order without assuming a prespecified model. Based on the corresponding design, we are able to find the optimal order. As such, we provide a way to explore the ordering problem.

To demonstrate the usefulness of the proposed method, we assess the D-efficiencies of the constructed designs under existing models, such as the PWO model (van Nostrand (1995); Voelkel (2019)), CP model (Yang, Sun and Xu (2021)), tapering PWO model (Peng, Mukerjee and Lin (2019)), first-order PWOD model (6.4), second-order PWOD model (6.5), second-order PWO model (Mee (2020)), first-order flexible position (FP) model (Stokes and Xu (2022), quadratic FP model (Stokes and Xu (2022)), and second-order FP model (Stokes and Xu (2022)). For simplicity, the case of $m = 4$ for $n = 6, 9, 12, 15, 18, 21$ is displayed in Table 4. It is shown that the proposed method leads to optimal and highly

Table 4. The D-efficiencies (%) of our designs compared with those of Stokes and Xu (2022), shown in parentheses, under existing models for different run sizes.

Model	n=6	n=9	n=12	n=15	n=18	n=21
PWO model		81.54	90.88	93.71	96.74	98.58
			(90.88)	(91.93)	(95.02)	(97.05)
CP model			100.00	95.90	95.40	97.07
			(100.00)	(95.70)	(95.04)	(96.67)
tapering PWO model		74.11	82.59	90.45	95.76	98.52
			(82.59)	(88.07)	(94.21)	(96.98)
second-order PWO model				76.51	87.97	93.62
				(0)	(0)	(90.46)
first-order PWOD model	94.57	98.33	100.00	99.25	99.44	99.67
	(86.30)	(92.67)	(100.00)	(98.09)	(98.47)	(98.86)
second-order PWOD model		80.04	100.00	96.29	96.15	97.51
		(0)	(100.00)	(96.00)	(95.56)	(97.01)
first-order FP model	94.57	98.33	100.00	99.25	99.44	99.67
	(86.30)	(92.67)	(100.00)	(98.09)	(98.47)	(98.86)
quadratic FP model		91.56	100.00	97.41	95.75	98.58
		(79.45)	(100.00)	(96.81)	(96.86)	(97.81)
second-order FP model		80.04	100.00	96.29	96.15	97.51
			(100.00)	(96.00)	(95.56)	(97.01)

efficient designs (most have at least 95% D-efficiencies) under the existing models. Moreover, we compare our designs with those of Stokes and Xu (2022) under possible different models in Table 4. The D-efficiencies of the designs obtained by Stokes and Xu (2022) are listed in parentheses in Table 4. The results show that the D-efficiencies of the designs constructed using the proposed method are higher than those of Stokes and Xu (2022) in almost all cases.

When the run size of the constructed design is smaller than $\binom{m}{2}$, the design can be considered as a supersaturated design, even for the PWO model. The construction methods and corresponding modeling technique for such supersaturated designs (see Lin (1993)) for OofA experiments are left to future research. Furthermore, we consider only the distance between each pair of components. One can also measure the similarity between the fractional designs and the full design based on the t-tuples ($t > 2$) sub-array of the PWOD arrays. This is also left to future research. It is anticipated that our results will provide a fresh viewpoint on investigating the ordering problem.

Supplementary Material

The online Supplementary Material includes the proofs of the propositions and theorems, a comparison with existing results, optimal designs under the D-efficiency and $\chi^2_P(D)$ criteria, and additional details on the case studies with the OofA experiment and the job scheduling problem.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (11871288, 11771220 and 12101357), National Ten-Thousand Talents Program, Natural Science Foundation of Tianjin (19JCZDJC31100), National Science Foundation (DMS-18102925), and the PhD Candidate Research Innovation Fund of Nankai University. The authors would like to thank Min-Qian Liu and Gennian Ge for their valuable comments. The authorship is listed in alphabetic order.

References

Ai, M. Y., Fang, K. T. and He, S. Y. (2007). $E(\chi^2)$-optimal mixed-level supersaturated designs. *Journal of Statistical Planning and Inference* **137**, 306–316.

Chen, J., Mukerjee, R. and Lin, D. K. J. (2020). Construction of optimal fractional Order-of-Addition designs via block designs. *Statistics Probability Letters* **161**, 108728.

Chen, J., Peng, J. and Lin, D. K. J. (2021). A statistical perspective on non-deterministic polynomial-time hard ordering problems: Making use of design for Order-of-Addition experiments. *Computer & Industrial Engineering* **162**, 107773.

Ding, X., Matsuo, K., Xu, L., Yang, J. and Zheng, L. (2015). Optimized combinations of bortezomib, camptothecin, and doxorubicin show increased efficacy and reduced toxicity in treating oral cancer. *Anti-Cancer Drugs* **26**, 547–554.

Fang, K., Ge, G. and Liu, M. (2002). Uniform supersaturated design and its construction. *Science in China Series A: Mathematics* **45**, 1080–1088.

Fang, K.-T., Lin, D. K. J., Liu, M.-Q. (2003). Optimal mixed-level supersaturated design. *Metrika* **58**, 279–291.

Fuleki, T. and Francis, F. J. (1968). Quantitative methods for anthocyanins. 3. Purification of cranberry anthocyanins. *Journal of Food Science* **33**, 266–274.

Hermelin, D., Kubitza, J.-M., Shabtay, D., Talmon, N. and Woeginger, G. J. (2019). Scheduling two agents on a single machine: A parameterized analysis of NP-hard problems. *Omega* **83**, 275–286.

Karim, M., McCormick, K. and Kappagoda, C. T. (2000). Effects of cocoa extracts on endothelium-dependent relaxation. *The Journal of Nutrition* **130**, 2105S–2108S.

Leung, J. Y.-T. (2004). *Handbook of Scheduling: Algorithms, Models and Performance Analysis*. Chapman and Hall, London.

Lin, D. K. J. (1993). A new class of supersaturated designs. *Technometrics* **35**, 28–31.

Lin, D. K. J. and Peng, J. (2019). Order-of-addition experiments: A review and some new thoughts (with discussion). *Quality Engineering* **31**, 49–59.

Mee, R. W. (2020). Order of addition modeling. *Statistica Sinica* **30**, 1543–1559.

Pinedo, M. L. (2016). *Scheduling: Theory, Algorithms, and Systems*. 5nd Edition. Springer, New York.

Peng, J., Mukerjee, R. and Lin, D. K. J. (2019). Design of order-of-addition experiments. *Biometrika* **106**, 683–694.

Shinohara, A. and Ogawa, T. (1998). Stimulation by Rad52 of yeast Rad51-mediated recombination. *Nature* **391**, 404–407.

Stokes, Z. and Xu, H. Q. (2022). A position-based approach for design and analysis of order-of-addition experiments. *Statistica Sinica* **32**, 1467–1488

Townsend, W. (1978). The single machine problem with quadratic penalty function of completion times: A branch-and-bound solution. *Management Science* **24**, 530–534.

van Greevenbroek, K. and Jedwab, J. (2018). A new structure for difference matrices over abelian p-groups. *arXiv:1806.04279v2*.

van Nostrand, R. C. (1995). Design of experiments where the order of addition is important. In *Proceedings of the Section on Physical and Engineering Sciences*, 155–160. The Association, Alexandria.

Voelkel, J. G. (2019). The design of order-of-addition experiments. *Journal of Quality Technology* **51**, 230–241.

Wei, W. (2019). Single machine scheduling with stochastically dependent times. *Journal of Scheduling* **22**, 677–689.

Wiens, D. P. (1991). Designs for approximately linear regression: Two optimality properties of uniform design. *Statistics & Probability Letters* **12**, 217–221.

Winker, P., Chen, J. and Lin, D. K. J. (2020). The construction of optimal design for order-of-addition experiment via threshold accepting. In *Contemporary Experimental Design, Multivariate Analysis and Data Mining* (Edited by J. Fan and J. Pan), 93–109. Springer, Cham.

Yamada, S. and Lin, D. K.J. (1999). Three-level supersaturated designs. *Statistics & Probability Letters* **45**, 31–39.

Yang, J.-F., Sun, F. and Xu, H. (2021). A component-position model, analysis and design for order-of-addition experiments. *Technometrics* **63**, 212–224.

Yin, J. (2003). Constructions of difference covering arrays. *Journal of Combinatorial Theory, Series A* **104**, 327–339.

Zhao, Y., Lin, D. K. J. and Liu, M.-Q. (2021). Designs for order-of-addition experiments. *Journal of Applied Statistics* **48**, 1475–1495.

Jianbin Chen

Department of Statistics, Purdue University, West Lafayette, IN 47907, USA.

E-mail: chenjianbinlzu@163.com

Xiaoxue Han

School of Statistics, Qufu Normal University, Qufu, Shandong, 273165, China.

E-mail: hanxiaoxue1229@163.com

Dennis K.J. Lin
Department of Statistics, Purdue University, West Lafayette, IN 47907, USA.
E-mail: dkjlin@purdue.edu
Liuqing Yang
School of Statistics and Data Science, LPMC & KLMDASR, Nankai University, Tianjin 300071, China.
E-mail: yliuqing0714@163.com
Yongdao Zhou
School of Statistics and Data Science, LPMC & KLMDASR, Nankai University, Tianjin 300071, China.
E-mail: ydzhou@nankai.edu.cn

(Received April 2021; accepted November 2021)