Statistica Sinica Preprint No: SS-2021-0410	
Title	Designs for Order-of-Addition Screening Experiments
Manuscript ID	SS-2021-0410
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202021.0410
Complete List of Authors	Zack Stokes and Hongquan Xu
Corresponding Authors	Hongquan Xu
E-mails	hqxu@stat.ucla.edu
Notice: Accepted version subject to English editing.	

Statistica Sinica

Designs for Order-of-Addition Screening Experiments

Zack Stokes and Hongquan Xu

University of California, Los Angeles

Abstract: When studying the relationship between the order of a set of components and a measured response in an order-of-addition experiment, the number of components may exceed the number of available positions. In this case there is an added layer of complexity in which the experimenter is tasked with locating both the best combination of components and its corresponding best order. Akin to the standard order-of-addition setup, the number of possible sequences grows quickly with the number of components, rendering a brute force approach unfeasible. This necessitates the development of parsimonious designs for these order-of-addition screening experiments. We present a framework for constructing optimal and near-optimal screening designs under adapted versions of two prominent order-of-addition models. We apply our order-of-addition screening designs to job scheduling problems with job rejection penalties in the context of both a single-shot experiment and an active learning framework for sequential experimentation. The proposed designs not only offer precise effect estimation and accurate predictions, but also facilitate quick convergence to the optimal ordering in sequential experiments.

Key words and phrases: Active learning, experimental design, job scheduling, optimal design, orthogonal array, screening experiment.

1. Introduction

In many physical and computer experiments, the order in which components are added to a mixture or steps are performed in a process can have a strong relationship with the response. However, as the number of components increases, the sample space suffers from a combinatorial explosion and performing all permutations becomes impossible. Accordingly, our goal is to design an order-of-addition experiment comprised of an informative and economical subset of all permutations that provides for robust estimation of model parameters.

Order-of-addition experiments have applications to many physical and simulated processes in a variety of fields from medical and food science to mechanics and engineering. Lin and Peng (2019) gives a review of these application areas, and Stokes and Xu (2022) provides many references that demonstrate the prevalence of order-of-addition effects in the context of drug combination problems. Another area in which order-of-addition problems are common is in the field of job sequencing and operations research through the flowshop permutation and traveling salesman problems (Xiao and Xu, 2021). However, despite the prevalence of these problems in many areas of research and application, rigorous statistical methods for designing such experiments are still in the early stages, especially for non-standard situations. This paper seeks to fill some of these gaps.

The study of statistical modeling for order-of-addition experiments can be broken into two primary groups. First is the class of relative-position effects models. Models in this class make the assumption that for a set of components it is the relative position of each component to the other components in the ordered sequence that impacts the response. The pairwise ordering (PWO) model is the primary model in this class (Van Nostrand, 1995). In this model, an indicator is generated for each pair of components such that it is assigned the value 1 if the components are in increasing order (i.e., component i appears before component j with $i < j$) and -1 otherwise. This class of models has been expanded further and many optimal and near-optimal designs have been constructed (Voelkel (2019), Peng et al. (2019), Zhao et al. (2021a), Chen et al. (2020), Schoen and Mee (2021)). Notably, Mee (2020) proposed a triplets ordering model which considers the ordering of the pairs of pairs (with one component overlapping) within each run.

The other class of order-of-addition models consists of models that make an absolute-position effects assumption. This means that the position value of each component is considered directly and assumed to have some relationship with the response. This class was first studied by Yang et al. (2021) under the component-position (CP) model in which an indicator variable is created for each component-position pair. The authors also constructed a class of optimal designs under this model, the component orthogonal array (COA). Stokes and Xu (2022) created a class of absolute-position effects models using orthogonal polynomials and proposed a COA-based construction for optimal and near-optimal designs under these models.

Despite the growing amount of literature on order-of-addition experiments, there are cases in which a researcher may have a larger pool of components than the number of available positions that the study can accommodate. For example, when working with a large collection of anti-tumor drugs, the practical administration of multiple drugs within a period of time may limit the maximum number of positions available in the order. In such a situation the experimenter needs to screen the drug combinations to determine which subset produces the best result, while also understanding the impact of the drug sequence on the response. The authors were consulted by a research team to design and analyze such an experiment. In this case a set of five anti-bacterial drugs are being tested for their ability to reduce cell bacteria count. However, physical limitations require that only three, and exactly three, of the five drugs be administered in each run in order to determine the optimal subset and corresponding ordering. Such a limitation necessitates the development of special designs. As another example, consider a traveling salesman problem in which the salesman must maximize profit and minimize cost by visiting a subset of available sites in a suitable order. This situation is encountered in single vehicle routing problems wherein balancing travel times with potential profits result in an optimal route that only visits a subset of sites (Bruni et al., 2019). To our knowledge this general problem has not yet been studied in the context of order-of-addition, yet the same combinatorial challenges that impede the standard order-of-addition problem, in which each component is used exactly once in every sequence, are present here. We aim to formalize this setup and develop efficient and robust designs for conducting such experiments.

The paper is outlined as follows. In Section 2, we make adjustments to two prominent order-of-addition models to capture the effects of the component subset and sequence on the measured response in the screening problem. Section 3 presents several design construction algorithms for choosing D-optimal subsets of the full design for different choices of the model, size of the component pool, and number of available positions by leveraging the properties of existing designs for the standard order-of-addition problem. To demonstrate the value of these designs, Section 4 applies our order-of-addition screening designs to job scheduling problems with job rejection penalties in the context of both a single-shot experiment and an active learning framework for sequential experimentation. We find that the proposed designs offer precise effect estimation and accurate predictions when treated as a single-shot experiment and fast convergence to the optimal ordering in sequential experiments. Section 5 concludes the work and all proofs are given in the Supplementary Materials.

2. Screening Models and Full Design Optimality

To align with the drug sequence application, we assume that the number of available positions in the order is fixed throughout the experiment and is represented by q with $1 < q < m$, where m is the total number of available components. In practice, q is determined by physical limitations of the application and our experiment aims to incorporate these constraints. The m components are denoted for convenience as $0, 1, \ldots, m-1$, arbitrarily. Under this setup there are a total of $\binom{m}{q}q! = \frac{m!}{(m-q)!}$ possible ways to assign the m components to the q positions. Each order-of-addition design is given in terms of a component matrix $\boldsymbol{A}$ in which each column $\boldsymbol{a}_j$ represents position j and a_{ij} gives the component added in position j of run i. We refer to the design which contains every possible subset/permutation pair as the full screening design, denoted by $\mathbf{S}_{m,q}$. For example, for $m = 5, q = 3$ the component matrix $\mathbf{S}_{5,3}$ has 60 runs or sequences. As the number of components increases, the number of possible sequences grows beyond what can be afforded by a single experiment. With no existing methodology to cover this case, our aim is to construct efficient order-of-addition screening designs that include only a small fraction of the total set of sequences.

To analyze data from screening experiments, we consider two existing order-of-addition models and modify them to accommodate the screening problem. First is the CP model from Yang et al. (2021). For the standard order-of-addition problem the authors constructed an indicator $z_{kj}^{(i)}$ for each component-position pair (k, j) such that $z_{kj}^{(i)}$ is 1 if $a_{ij} = k$ and 0 otherwise. Typically the constraint $\sum_{k=1}^{m} z_{kj}^{(i)} = 1$ for any i and j necessitates that we remove the terms associated with one component and one position. However, in our case each run is a permutation of at most $m - 1$ distinct components instead of a full permutation of all m components. Thus, we do not need to remove the terms associated with one of the positions to make the model estimable. The component-position screening (CPS) model is then

$$y = \gamma_0 + \sum_{k=1}^{m-1} \sum_{j=1}^{q} z_{kj}\gamma_{kj} + \varepsilon, \tag{2.1}$$

where y is the response, γ_0 is the intercept, z_{kj} is an indicator for the component-position pair (k, j), γ_{kj} is the parameter representing the effect of component k being added at the j^{th} position, and ε is a normal random error with mean 0 and constant variance σ^2. All errors are assumed to be independent. Along with the standard CP model, Yang et al. (2021) introduced a class of optimal designs, the Component Orthogonal Array (COA). These designs are level balanced and have the property that every pair of components shows up equally often in any two-column sub-array. These designs will be one of the building blocks for our order-of-addition screening designs.

We also consider the PWO model first introduced by Van Nostrand (1995) and Voelkel (2019). In this model a set of pseudo-factors $\{I_{ij},\ 0 \leq i < j \leq m\}$ is created such that each corresponds to the pairwise ordering of the components. In the standard model, each factor I_{ij} has two levels, 1 and -1, indicating whether or not component i is added before component j. However, in the screening case not every component is present in the sequence, so for each I_{ij} in which component i or j is missing from the sequence we assign a value of 0. With this change, the pairwise ordering screening (PWOS) model is given by

$$y = \beta_0 + \sum_{i<j} \beta_{ij} I_{ij} + \varepsilon, \tag{2.2}$$

with random error $\varepsilon \sim N(0, \sigma^2)$.

In addition to this version of the model, we will use the Order-of-Addition Orthogonal Array (OofA-OA) class of designs that was proposed by Voelkel and Gallagher (2019) and has since been expanded by Mee (2020) and Schoen and Mee (2021). Each OofA-OA in m components with n runs has the property that each pair of pseudo-factors (I_{ij}, I_{kl}) meets the following conditions:

1. If $i \neq k$, $i \neq l$, $j \neq k$, and $j \neq l$, the factors are orthogonal.

2. If $i = k$ or $j = l$, the inner product of the factors is $n/3$.

3. If $i = l$ or $j = k$, the inner product of the factors is $-n/3$.

The inner product of the factors is the sum of the product of I_{ij} and I_{kl} over all n runs. These designs will provide a basis for the construction of order-of-addition screening designs under the PWOS model.

Note that while we have relabeled the CP and PWO models as CPS and PWOS, respectively, this is done primarily to differentiate the study of this problem from that of the standard order-of-addition problem. The fundamental structure of each model is largely unchanged up to the minor alterations required to accommodate the data from component screening experiments. It remains the subject of future research to consider other modeling approaches (*e.g.*, position-based models, Gaussian Process models, etc.), as well as the feasibility of a two-stage experiment in which we first screen the m components and then use the remaining q components in a standard order-of-addition experiment.

Under the CPS and PWOS models, the designs we construct can be evaluated using the popular D-optimality criterion. For an n-run design $\boldsymbol{\xi} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, let $\boldsymbol{X} = (\boldsymbol{f}(\boldsymbol{x}_1), \boldsymbol{f}(\boldsymbol{x}_2), \ldots, \boldsymbol{f}(\boldsymbol{x}_n))^{\mathrm{T}}$ be the model matrix of the chosen linear model $\boldsymbol{f}$, and $\boldsymbol{M}(\boldsymbol{\xi}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}/n$ be the per-run information matrix. A D-optimal design maximizes $|\boldsymbol{M}(\boldsymbol{\xi})|$. The D-optimality criterion seeks to minimize the volume of the confidence ellipsoid for the parameter estimates. This optimality can be verified using the general equivalence theorem (Silvey, 1980) for approximate designs where we determine the proportion of the observations at each design point instead of the number of replicates. With this in mind, the first step in finding smaller designs for order-of-addition screening experiments is to show that the full design is optimal for the two models discussed above, so that we may use it as

$m(q)$	4(3)	5(3)	5(4)	6(3)	6(4)	6(5)	7(3)	7(4)	7(5)	7(6)
CPS	10	13	17	16	21	26	19	25	31	37
PWOS	7	11	11	16	16	16	22	22	22	22

Table 1: Number of parameters (p) in the CPS and PWOS models for different values of m and q.

a reference design for future designs. To this end we have the following results:

Theorem 1. *The full design* $\mathbf{S}_{m,q}$ *is D-optimal for the CPS model (2.1) with* $m \geq 3$ *and* $1 < q < m$.

Theorem 2. *The full design* $\mathbf{S}_{m,q}$ *is D-optimal for the PWOS model (2.2) with* $m \geq 3$ *and* $1 < q < m$.

While this design is not practically useful, and these results are not surprising, the validation of the full design's D-optimality allows us to benchmark the quality of any proposed design against this optimal one. For convenience, we define the D-efficiency of a design $\boldsymbol{\xi}$ under the chosen model relative to $\mathbf{S}_{m,q}$ as $D(\boldsymbol{\xi}) = \{|\boldsymbol{M}(\boldsymbol{\xi})|/|\boldsymbol{M}(\mathbf{S}_{m,q})|\}^{1/p}$, where p is the number of columns of the model matrix $\boldsymbol{X}$. Generally, we would prefer a design with a run size close to that of the number of columns in the model matrix. This value for different choices of m, q and model are given in Table 1. With these preliminary steps complete, we are ready to construct smaller, optimal and near-optimal order-of-addition screening designs.

3. Component Screening Design Constructions

Considering the two models described in the previous section, we offer two primary design constructions and a third construction for a special case not covered by the other two. The two primary constructions are built upon existing order-of-addition designs. To motivate the construction of order-of-addition screening designs for the CPS model, Algorithm 1 utilizes the flexible construction of COA-based designs proposed in Stokes and Xu (2022). For the PWOS model, Algorithm 2 considers the special case of $q = 3$ with even m while Alogrithm 3 takes advantage of the properties of the OofA-OAs constructed in Schoen and Mee (2021) to cover the remaining cases. Each of these constructions produces fractional designs for various m and q that are D-optimal under one or both models. For each method, we establish settings of m, q and n under which the resulting design is D-optimal. To understand the robustness properties of our designs to model misspecification, we explore the efficiency of the designs produced for one model under the other.

3.1 Optimal Design Construction for the CPS Model

Stokes and Xu (2022) showed that their designs for the standard order-of-addition problem, denoted by $\mathbf{F}_{n,m}$ for n runs in m components, can achieve high efficiency on the standard CP and PWO models with minor tuning. We base our construction of efficient designs for the CPS model on these designs. To understand what is to follow, it may be helpful to review the construction method in Stokes and Xu (2022), which has been reproduced in the Supplementary Materials. Using the $\mathbf{F}_{n,m}$ designs, we propose the following algorithm for constructing order-of-addition screening designs with a pool of m components, n runs and q positions. Let $\lceil x \rceil$ be the smallest integer that is equal to or larger than x.

Algorithm 1.

Step 1. Generate the $n \times m$ matrix $\mathbf{F}_{n,m}$ using the first five steps of the algorithm from Stokes and Xu (2022).

Step 2. Construct an $n \times q$ matrix $\mathbf{S}^{\text{cp}}_{n,m,q}$ by taking the first q odd-numbered columns of $\mathbf{F}_{n,m}$ if $q \leq m/2$. Otherwise take the $\lceil m/2 \rceil$ odd-numbered columns followed by the first $q - \lceil m/2 \rceil$ even-numbered columns.

Step 3. Permute the columns of $\mathbf{S}^{\text{cp}}_{n,m,q}$ to improve its performance under a chosen criterion.

In the development of Algorithm 1 we have found that the choice of which columns of the design to take in Step 2 does not affect the endpoint efficiency of the design under the CPS model. However, taking the odd-numbered columns of $\mathbf{F}_{n,m}$ first followed by the even-numbered columns produces pairwise pseudo-factors in the PWOS model with better balance properties, and in turn yields much higher D-efficiency, than taking the first or last q columns or taking a random subset of columns. For small m and q, a complete search over all $\binom{m}{q}$ sub-designs to maximize the chosen criterion could be beneficial.

To illustrate the construction we consider the case $m = 5, q = 3$ for which the full design $\mathbf{S}_{5,3}$ has 60 runs. Instead of using this full design, we construct a design in 20 runs by first generating the design $\mathbf{F}_{20,5}$ (Table 2a), then taking the three odd-numbered columns, and finally permuting the columns to maximize the efficiency under the PWOS model (Table 2b). This design is D-optimal under the CPS model and has high efficiency relative to $\mathbf{S}_{5,3}$ under the PWOS model (approximately 0.91).

Figure 1 shows the performance of our algorithm for the PWOS and CPS models under three settings of m and q: $(m = 5, q = 3)$, $(m = 7, q = 3)$, and $(m = 7, q = 4)$. It is important to note that the order-of-addition design $\mathbf{F}_{n,m}$ can only be constructed when m is prime or a prime power. This is a limitation that could be addressed by using the recursive construction of COA designs with a non-prime number of components presented in Huang

Table 2: (a) 20-run optimal order-of-addition design with $m = 5$, $\mathbf{F}_{20,5.}$ (b) 20-run D-optimal screening design under the CPS model with $m = 5, q = 3$, $\mathbf{S}^{\text{cp}}_{20,5,3}$, generated from Algorithm 1.

(a)

Run	a_1	a_2	a_3	a_4	a_5
1	0	1	2	3	4
2	1	2	3	4	0
3	2	3	4	0	1
4	3	4	0	1	2
5	4	0	1	2	3
6	0	2	4	1	3
7	1	3	0	2	4
8	2	4	1	3	0
9	3	0	2	4	1
10	4	1	3	0	2
11	0	3	1	4	2
12	1	4	2	0	3
13	2	0	3	1	4
14	3	1	4	2	0
15	4	2	0	3	1
16	0	4	3	2	1
17	1	0	4	3	2
18	2	1	0	4	3
19	3	2	1	0	4
20	4	3	2	1	0

(b)

Run	a_1	a_2	a_3
1	2	0	4
2	3	1	0
3	4	2	1
4	0	3	2
5	1	4	3
6	4	0	3
7	0	1	4
8	1	2	0
9	2	3	1
10	3	4	2
11	1	0	2
12	2	1	3
13	3	2	4
14	4	3	0
15	0	4	1
16	3	0	1
17	4	1	2
18	0	2	3
19	1	3	4
20	2	4	0

(2021) in Step 1 of Algorithm 1.

We observe from Figure 1 that the construction yields designs with high D-efficiency relative to the full screening design $\mathbf{S}_{m,q}$ for both models. Under the PWOS model our designs perform well, achieving efficiency over 0.90 in most cases where the run size is suitable for estimating the model. However, our algorithm was unable to find the optimal design under the

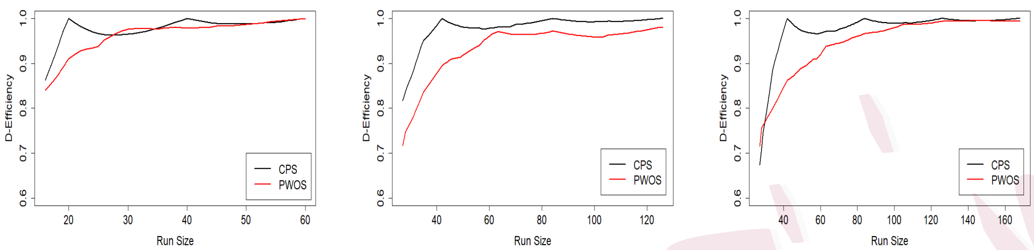

Figure 1: D-efficiency of designs $\mathbf{S}^{\text{cp}}_{n,m,q}$ relative to the full design $\mathbf{S}_{m,q}$ generated for various run sizes with (left) $m = 5, q = 3$, (middle) $m = 7, q = 3$, and (right) $m = 7, q = 4$.

PWOS model for any case. This is not surprising as the $\mathbf{F}_{n,m}$ designs used to generate these designs are similarly only near-optimal under the standard PWO model. On the other hand, the generated designs under the CPS model are either optimal or highly efficient. More specifically, we have the following general result about designs generated from Algorithm 1 for the CPS model.

Theorem 3. *For $n \leq m!$ and $1 < q < m$ with n divisible by $m(m-1)$ and m a prime or prime power, $\mathbf{S}^{\text{cp}}_{n,m,q}$ is D-optimal under the CPS model.*

These results indicate that for all possible values of q, our construction algorithm can generate optimal designs under the CPS model with affordable run sizes. We also observe from Figure 1 that our designs are fairly robust to violations of the assumption of whether absolute or relative position effects are more suitable for capturing the true relationship between the order and the response.

3.2 Optimal Design Constructions for the PWOS Model

While the construction presented in Algorithm 1 produces near-optimal designs under the PWOS model, none of the resulting designs are D-optimal. To fill this gap we propose two separate constructions that can produce fractional, D-optimal designs under this model. For $q \geq 4$ our designs are constructed from OofA-OAs, such as those from Voelkel and Gallagher (2019) and Schoen and Mee (2021). However, for $q < 4$ a separate construction is required. The following result establishes the basis of our method for constructing D-optimal designs under the PWOS model with $q = 3$.

Theorem 4. *For $m > 2$, the run size n of any design with D-efficiency of 1 relative to the full design under the PWOS model has the following constraints:*

(i) If $q = 2$, the minimum run size of a design with D-efficiency 1 is $n = 2\binom{m}{2}$.

(ii) If $q = 3$ and m is odd, the minimum run size of a design with D-efficiency 1 is $n = 6\binom{m}{3}$.

(iii) If $q = 3$ and m is even, the only run size for which a design with D-efficiency 1 exists with $n < 6\binom{m}{3}$ is $n = 3\binom{m}{3}$.

The implication of the above result is that there is no fractional design with D-efficiency 1 when $q = 2$, or when $q = 3$ and m is odd. For $q < 4$ this leaves only the case that $q = 3$ with even m. In this case only a half-fraction optimal design exists. For any even m, the following algorithm generates this design along with an efficient design for any $n < 3\binom{m}{3}$.

Algorithm 2.

Step 1. Generate the set of $\binom{m}{3}$ 3-component combinations (i, j, k) with $0 \leq i < j < k \leq m - 1$.

Step 2. For every 3-component combination such that $i + j + k$ is even, construct the 3×3 matrix $\boldsymbol{D}_{ijk}$ given by

$$\boldsymbol{D}_{ijk} = \begin{bmatrix} i & k & j \\ j & i & k \\ k & j & i \end{bmatrix}.$$

Step 3. For every 3-component combination such that $i + j + k$ is odd, construct the 3×3 matrix $\boldsymbol{D}_{ijk}$ given by

$$\boldsymbol{D}_{ijk} = \begin{bmatrix} i & j & k \\ j & k & i \\ k & i & j \end{bmatrix}.$$

Step 4. Construct $\mathbf{S}^{\text{pwo}}_{n,m,3}$ by first row-wise concatenating the $\boldsymbol{D}_{ijk}$ for all $0 \leq i < j < k \leq m-1$ such that $i+j+k$ is even, then concatenating

Table 3: 12-run, half-fraction D-optimal screening design for the PWOS model with $m = 4, q = 3$, $\mathbf{S}^{\text{pwo}}_{12,4,3}$, produced by Algorithm 2.

Run		a_1	a_2	a_3
1		0	3	1
2	$\boldsymbol{D}_{013}$	1	0	3
3		3	1	0
4		1	3	2
5	$\boldsymbol{D}_{123}$	2	1	3
6		3	2	1
7		0	1	2
8	$\boldsymbol{D}_{012}$	1	2	0
9		2	0	1
10		0	2	3
11	$\boldsymbol{D}_{023}$	2	3	0
12		3	0	2

the remaining $\boldsymbol{D}_{ijk}$ and taking the first n rows.

Table 3 demonstrates this construction method for the case of $m = 4$ and $q = 3$. For this scenario there are four 3-component combinations, with $(0, 1, 3)$ and $(1, 2, 3)$ summing to an even number, and $(0, 1, 2)$ and $(0, 2, 3)$ summing to an odd number. Concatenating the respective $\boldsymbol{D}_{ijk}$ for each of these combinations produces $\mathbf{S}^{\text{pwo}}_{12,4,3}$, a 12-run D-optimal design for the PWOS model, a half-fraction of the 24-run full design.

Algorithm 2 covers the special case of generating half-fraction screening designs with $q = 3$. We note that the construction as presented can be used to generate designs for odd m, but the result will only achieve high efficiency, not optimality, as this case is not covered in the following theorem.

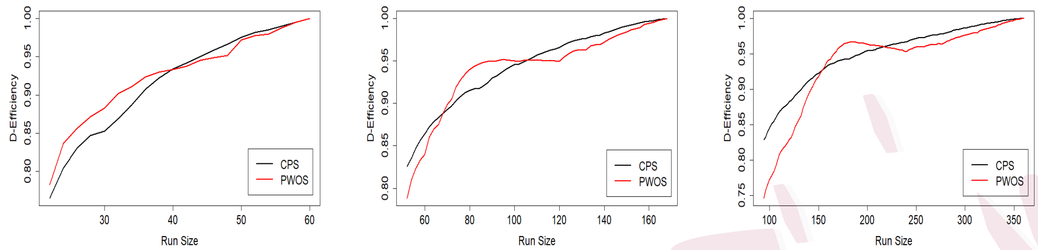

Figure 2: D-efficiency of designs $\mathbf{S}^{\text{pwo}}_{n,m,3}$ relative to the full design $\mathbf{S}_{m,3}$ generated for a range of run sizes with (left) $m = 6$, (middle) $m = 8$, and (right) $m = 10$.

Theorem 5. *For $n = 3\binom{m}{3}$ and m even, $\mathbf{S}^{\text{pwo}}_{n,m,3}$ is D-optimal under the CPS and PWOS models.*

To visualize the performance of these designs we consider the cases $(m = 6, q = 3)$, $(m = 8, q = 3)$, and $(m = 10, q = 3)$ in Figure 2. We observe from this figure that the designs generated by Algorithm 2 achieve high D-efficiency across both models for all run sizes considered, with the half fraction design having $n = 3\binom{m}{3}$ runs being D-optimal as determined in Theorem 5. In addition to providing efficient designs for even m with $q = 3$ under the PWOS model, this construction provides optimal designs under the CPS model for several non-prime or non-prime power values of m, filling another gap of Algorithm 1.

For $q \geq 4$ we instead use OofA-OAs in q components as the building blocks of D-optimal designs, leveraging the special properties of these designs detailed in Section 2. From Voelkel and Gallagher (2019) we know that optimal OofA-OAs must have a run size that is divisible by 12. Thus, for a given q, the smallest optimal order-of-addition design has $N = 12\lceil (\binom{q}{2} + 1)/12 \rceil$ runs. With these properties in mind, Algorithm 3 generates optimal or near optimal designs for any $n \leq N\binom{m}{q}$ with $m > 4$ and $3 < q < m$.

Algorithm 3.

Step 1. Construct $\boldsymbol{OA}_{N,q}$, the smallest optimal OofA-OA in q components $\{0, 1, \ldots, q-1\}$, with $N = 12\lceil (\binom{q}{2} + 1)/12 \rceil$ runs.

Step 2. Generate the set of $\binom{m}{q}$ q-component combinations $(i_1, i_2, \ldots, i_q)$ with $0 \leq i_1 < i_2 < \ldots < i_q \leq m - 1$.

Step 3. For every q-component combination create the $N \times q$ matrix $\boldsymbol{OA}_{N,q,i_1 i_2 \ldots i_q}$ by substituting the levels of $\boldsymbol{OA}_{N,q}$ according to the permutation

$$\begin{pmatrix} 0 & 1 & \ldots & q-1 \\ i_1 & i_2 & \ldots & i_q \end{pmatrix}.$$

Step 4. Construct $\mathbf{S}^{\text{pwo}}_{m,q}$ by row-wise concatenating the $\boldsymbol{OA}_{N,q,i_1 i_2 \ldots i_q}$ for all $0 \leq i_1 < i_2 < \ldots < i_q \leq m - 1$.

Step 5. Generate $\mathbf{S}^{\text{pwo}}_{n,m,q}$ by selecting the rows of $\mathbf{S}^{\text{pwo}}_{m,q}$ sequentially. Specifically, we select the first run from the first $\boldsymbol{OA}$, the second run from the second $\boldsymbol{OA}$, and so on. Run $\binom{m}{q}+1$ is taken to be the second run of the first $\boldsymbol{OA}$ and run $\binom{m}{q}+2$ is taken to be the third run of the second $\boldsymbol{OA}$, etc. This process continues until n runs are selected for $\mathbf{S}^{\text{pwo}}_{n,m,q}$.

With this construction we can generate optimal designs for any combination of m and q. Specifically, Table 6 in the Supplementary Materials demonstrates the process of creating a 72-run design for the case $m = 6, q = 5$. We start with the 12-run order-of-addition design in 5 components given in Schoen and Mee (2021). This is represented as $\mathbf{OA}_{12,5,01234}$ in the left panel of Table 6. Next, we consider the five other 5-component combinations and substitute the levels according to the permutation given in Step 3 to create each $\mathbf{OA}_{12,5,i_1 i_2 \ldots i_5}$. After concatenating these six arrays we rearrange the rows as described in Step 5 to ensure a sufficient amount of information is present in the first n rows if n is less than the size of the complete design. We do this by selecting the first run from $\mathbf{OA}_{12,5,01234}$, followed by the second run from $\mathbf{OA}_{12,5,01235}$, and so on, cycling through runs 1 through 12 and each $\mathbf{OA}_{12,5,j}$ for $j = 1, \ldots, 6$ until all runs are accounted for. The result is a 72-run D-optimal design that is the one-fifth fraction of the full design with 360 runs given in Table 6 of the Supplementary Materials. We then have the following result.

Theorem 6. *For* $n = 12\lceil(\binom{q}{2}+1)/12\rceil\binom{m}{q}$, $\mathbf{S}^{\text{pwo}}_{n,m,q}$ *is D-optimal under the PWOS model.*

Following the results of Theorems 5 and 6, we achieve significant savings in terms of the number of runs compared to the full design $\mathbf{S}_{m,q}$ by using the screening designs produced by Algorithms 2 and 3. To further demonstrate the efficiency of these designs, Figure 6 in the Supplementary Materials shows the D-efficiency of the designs $\mathbf{S}^{\text{pwo}}_{n,m,q}$ under the PWO model for several values of m and q. In general, we find that once the run size is sufficiently large enough to estimate the PWOS model, the design after row rearrangement is quite efficient, with D-efficiency greater than 0.80. From these results we conclude that between the three construction methods, we can generate designs that are efficient, parsimonious, and in some cases optimal for estimating one or both of the screening models. Our goal is now to see how researchers can use these designs in practice through some simulated order-of-addition screening experiments.

4. Order-of-Addition Screening Experiments in Practice

In order to demonstrate the value of our proposed designs in practice, we consider a collection of job scheduling problems of varying complexity. Job scheduling problems are a class of well known NP-hard problems that have been critically studied in operations research (Garey et al., 1976). Specifically, we borrow the setup from Zhao et al. (2021a) and consider a single machine which is tasked with sequentially processing jobs, each of which takes some fixed time to complete and requires some fixed costs to perform. However, our goal is to choose only q of the m available jobs to complete in a specific order such that a given response function is minimized, indicating that the sequence is in some sense the most efficient. This problem is common in high-volume manufacturing settings where processing all jobs is not possible due to inventory or time constraints (Shabtay et al., 2013, 2012; Zhong et al., 2014). Instead, a rejection penalty is added for each of the $m-q$ jobs that are not completed. For an ordered set $\boldsymbol{x} = \{x_1, \ldots, x_q\}$ of q jobs, the response we choose is the sum of a quadratic function (Townsend, 1978) and job rejection penalties given by

$$y(\boldsymbol{x}) = \sum_{i=1}^{q} c_{x_i} (\sum_{j=1}^{i} t_{x_j})^2 + \sum_{k \notin \boldsymbol{x}} p_k,$$

where t_{x_j} and c_{x_i} are the processing time of job x_j and the cost of job x_i, respectively, and p_k is the penalty for not completing job k. The penalty should be in the same unit (such as dollar amount) as the cost.

Table 4: Job scheduling matrices for $m = 4$ and $m = 6$.

Job	0	1	2	3
Time t	3	5	6	4
Cost c	7	3	2	6
Penalty p	90	85	100	80

Job	0	1	2	3	4	5
Time t	8	16	10	9	12	14
Cost c	16	5	12	13	9	7
Penalty p	107	98	110	89	96	101

Due to this ability to reject jobs, an assumed value of q may not produce the true optimal sequence. For these examples, we assume that external factors fix the number of jobs required in each sequence to be q. Alternatives to this include composite designs with sequences of varying length, a topic discussed further in Section 5. Considering this task, we demonstrate the value that our proposed order-of-addition screening designs provide in capturing the relationship between the job sequence and the endpoint penalty as well as in cheaply and efficiently uncovering the optimal job sequence.

4.1 Screening Experiments for Job Scheduling Problems

To evaluate how the proposed designs can be used to model the relationship between the component selection and sequence, we consider two situations ($m = 4$, $q = 3$ and $m = 6$, $q = 4$) similar to the problems considered in Zhao et al. (2021a). The scheduling matrices for these problems are given in Table 4.

For the four-job problem, we consider two designs which we have found are D-optimal for the respective models. For each design, the vector of

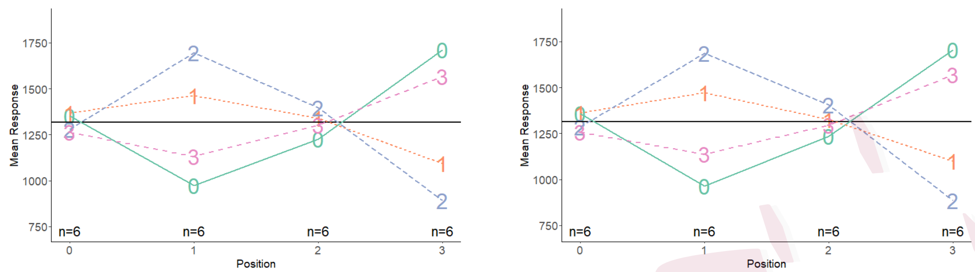

Figure 3: Component-position effects plots for the true data (left) and CPS predictions based on a 12-run D-optimal design (right) for the job scheduling problem with $m = 4$, $q = 3$.

responses is calculated for each job sequence outlined by the runs of the design. We use the 12-run design from Algorithm 1 for training the CPS model and the 12-run design from Algorithm 2 for training the PWOS model. The CPS model achieves marginally better performance than the PWOS model in terms of fit ($R^2 = 0.99$ versus $R^2 = 0.98$, AIC $= 131.95$ versus AIC $= 147.78$), but includes more parameters (df $= 2$ versus df $= 5$). To test the predictive performance of each model we predict the response for each of the 24 sequences and measure the correlation of these predictions against the 24 true responses. The CPS model outperforms the PWOS model with a correlation of 0.99 compared to 0.97 for the PWOS model.

We can also visually interpret these models and draw some preliminary conclusions from the component-position effects plots in Figure 3. In each of these plots, the horizontal axis denotes the position in which the job is performed, and the vertical axis denotes the mean response, in this case the total expense of the job sequence including time, cost and rejection penalty. Each point denotes the mean response of all runs in which the labeled job is performed in the fixed position. For each job, the q dots corresponding to q different positions are connected to visualize the trend as that job is shifted to a later position in the sequence. The points at position 0 represent the mean response when the labeled component is omitted from the sequence. The number of observations used to compute each average is given along the x-axis. The solid horizontal line, used as a reference, represents the average response of all observations.

Figure 3 shows the component-position effects plots for the true data and for the CPS model predictions. Each plot is constructed from 24 observations. The left plot is built from the 24 true values and acts as a benchmark while the right plot is built from the 24 predictions of the fitted CPS model. We omit the plot of the PWOS model predictions to conserve space; however, we note that the overall interpretations are similar. Interpreting these plots we observe that the CPS model is adept at picking up the trends of the true data. If our aim is to minimize the total expense, then we should process job 0 first and job 2 third. The CPS model then indicates that we should process either job 1 or 3 second and not process the other. These interpretations align with the plot of the true data and the two sequences that obtain the smallest responses, $(0, 1, 2)$ and $(0, 3, 2)$. This visual analysis of course is only a first attempt at how researchers may use the proposed designs in concert with the screening models to draw substantive conclusions. Further study may be required to uncover and interpret any potential interactions between the components.

For the six-job problem we repeat this procedure using the 30-run D-optimal design from Algorithm 3 for the PWOS model and a design from Algorithm 1 that leverages the six-component COA from Huang (2021) for the CPS model. Training the models on their respective designs, we find that the CPS model demonstrates marginally greater performance, having df=9, $R^2 = 0.98$, AIC $= 494.86$, and correlation of 0.98 between predicted and true responses while the PWOS model uses fewer parameters with df=14, $R^2 = 0.98$, AIC $= 542.54$, and correlation of 0.94 between predicted and true responses. Both models are able to effectively capture the relationship between the sequence and the response even as the number of components increases. The full results of this study are given in the Supplementary Materials.

4.2 Sequential Screening Experiments for Job Scheduling

While we have seen that the proposed designs lead to stable, interpretable models, we have so far focused on single-shot designs where the entire budget of the experiment is used at once. However, in some cases it may be of interest to run a sequential experiment in which the goal is to find the best sequence as quickly as possible by first obtaining the responses from an initial design and then adding points to the design one at a time. We now aim to demonstrate the benefit of the proposed designs for this problem. For this procedure, we only consider the CPS model due to its advantage in prediction accuracy and use designs generated from Algorithm 1. We have also found that the choice of penalty has little impact on the CPS model whereas the performance of the PWOS model may deteriorate when the penalty varies substantially, so we simplify the approach by assuming that all jobs incur the same penalty if not included in the sequence. Specifically, we set job rejection penalty to 0.

We can now simulate the full job sequencing dataset for a fixed q and determine the sequence that produces the global minimum. Within this setup, we measure the benefit of choosing the designs we have constructed as the initial design over a random design as follows. To keep the number of runs in the experiment low while ensuring sufficient degrees of freedom for estimation, we set the number of initial runs to $n = q(m-1)+5$.

1. First collect the response from the design $\mathbf{S}^{\text{cp}}_{n,m,q}$ obtained from Algorithm 1 with $n = q(m-1)+5$ and record the minimum response.

2. Next fit the CPS model to the data and calculate the expected improvement (EI) for all remaining sequences in the pool (Jones et al., 1998). The EI for a given sequence $\boldsymbol{x}$ is calculated as

$$\text{EI}(\boldsymbol{x}) = (y^* - \hat{y}(\boldsymbol{x}))\Phi\Big(\frac{y^* - \hat{y}(\boldsymbol{x})}{\hat{\sigma}}\Big) + \hat{\sigma}\phi\Big(\frac{y^* - \hat{y}(\boldsymbol{x})}{\hat{\sigma}}\Big),$$

   where y^* is the minimum value observed so far, $\hat{y}(\boldsymbol{x})$ is the predicted response for input $\boldsymbol{x}$, ϕ and Φ are the probability density function and the cumulative density function of the standard normal distribution, respectively, and $\hat{\sigma}$ is the estimate of the standard error of the prediction. The EI value captures the model's uncertainty in the value of $y(\boldsymbol{x})$ by considering it as a realization of a normal distribution with mean $\hat{y}(\boldsymbol{x})$ and standard deviation $\hat{\sigma}$.

3. Add the design point with the largest EI value to the design and calculate its response.

4. Fit the CPS model again with the updated design and record the

Table 5: Job scheduling matrices for active learning sequential experiments with $m = 7$ (left) and $m = 11$ (right).

Job	0	1	2	3	4	5	6
Time t	6	1	11	1	2	21	2
Cost c	7	19	3	4	10	20	18

Job	0	1	2	3	4	5	6	7	8	9	10
Time t	6	27	13	11	20	20	5	10	20	21	17
Cost c	17	18	19	29	28	4	24	30	10	8	1

minimum response found so far.

5. Repeat 2-4 until the maximum number of iterations is reached.

To compare this approach to a random initial design, we repeat the process above with 100 random designs and average the resulting curves from tracking the minimum value in each iteration. We consider several job scheduling problems with $m = 7$ and $m = 11$, and q varying between 3 and 6. Table 5 gives the values of t and c for each job for the two problems. These values were uniformly sampled for each problem from $\{0, 1, \ldots, 3m\}$. Each combination of m and q presents a different job scheduling problem in which our goal is to find the optimal sequence of length q with as few runs as possible.

We first consider the problem with $m = 7$ jobs. We generate the designs $\mathbf{S}^{\text{cp}}_{n,7,q}$, where q takes the values $3, \ldots, 6$ and $n = q(m-1) + 5$. Applying each design in the sequential experimentation framework outlined above, the results for each value of q are presented in Figure 4. In these plots the dashed gray line represents the true global minimum. Each point on the

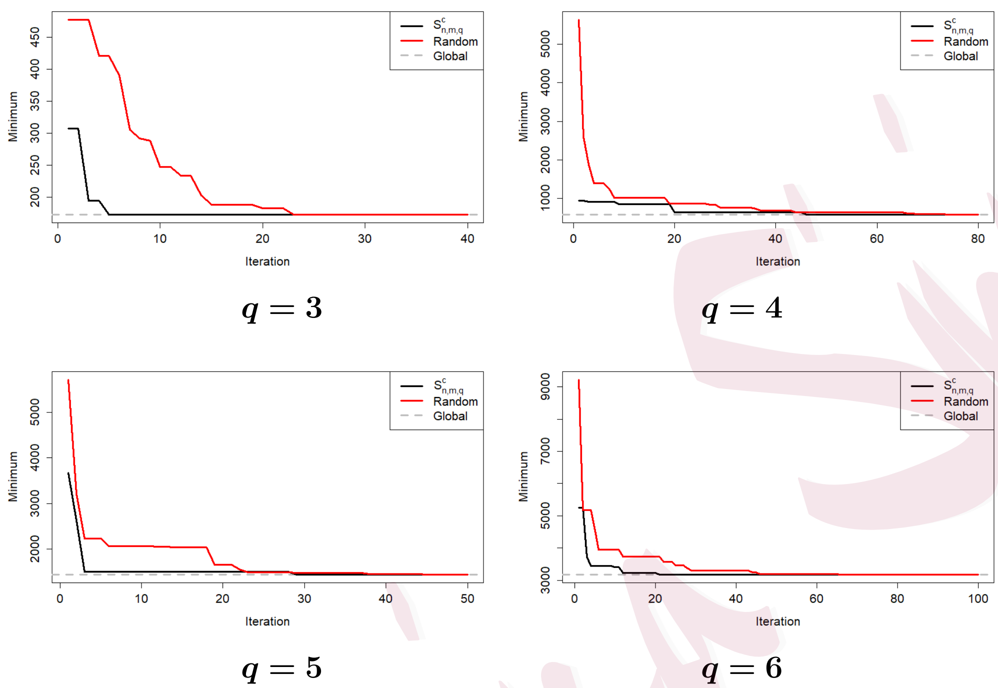

Figure 4: Convergence of the sequential job scheduling experiments to the true minimum response with $m = 7$ and $q = 3$–6 for different initial designs. The y-axis gives the minimum response observed up to the current iteration.

red curve represents the average minimum value obtained at the specified iteration across 100 random initial designs. Initializing the experiment with the design $\mathbf{S}^{\text{cp}}_{n,7,q}$ leads to convergence that is at least as fast as when starting from a typical random design. In fact, for most cases convergence is much faster under the proposed design. Specifically in the cases of $q = 3$ and $q = 6$, convergence to the global minimum occurs with less than half the number of iterations required for a typical random design. After accounting for the size of the initial designs this translates to roughly a 40% and 30%

reduction in the total budget required for the experiments, respectively.

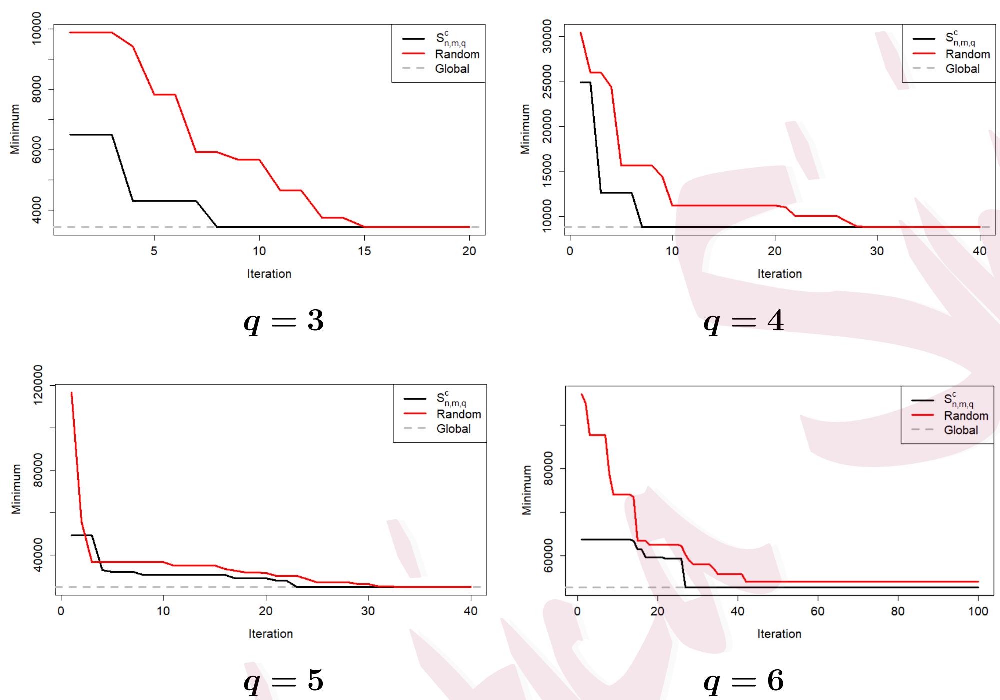

Figure 5: Convergence of the sequential job scheduling experiments to the true minimum response with $m = 11$ and $q = 3$–6 for different initial designs. The y-axis gives the minimum response observed up to the current iteration.

Considering the more difficult problem with $m = 11$ jobs, we again start from the $\mathbf{S}^{\text{cp}}_{n,11,q}$ designs for $q = 3, \ldots, 6$. The results of running the sequential experiment for each of these initial designs and 100 random designs are given in Figure 5. The true global minimum is again given as a dashed gray line in each figure. As with the simpler 7 job problem, for all situations the convergence of the algorithm when starting from the proposed design is at least as fast as when starting from a random design. In fact, in the four cases we have considered, the convergence is actually faster under our design, and in one case, $q = 6$, about 50% of the random designs are unable to converge with 100 additional runs. For the other three cases, this faster convergence leads to a substantial reduction in the total budget of the experiment, between roughly 9% and 27%.

We can gather from these two examples that starting from a design generated by Algorithm 1 often leads to much faster convergence than a random design, even as the total number of components or the value of q increases. In some cases the total budget required for the experiment when starting from this design is only a fraction of that under a randomly generated design of the same size. These examples demonstrate the potential value of our proposed designs. We could repeat this procedure with the optimal designs generated for the PWOS model by Algorithms 2 and 3, but as we have seen in the Section 4.1, both models do a good job of capturing the trends in the job sequencing simulated data, but the CPS model has better prediction accuracy.

In all of the problems considered here we also notice that the minimum achieved by $\mathbf{S}^{\text{cp}}_{n,m,q}$ before any additional points are added (iteration 0) tends to be much smaller than that of a typical random design. This is likely due to improved space-filling properties of the design $\mathbf{S}^{\mathrm{cp}}_{n,m,q}$, an observation that requires further investigation in future research.

5. Conclusion

We have studied the problem of designing order-of-addition experiments in cases where the number of components of interest m outnumbers the number of available positions in each sequence q. Like the standard order-of-addition problem in which each run is a permutation of all m components, the full design that includes all possible sequences grows too quickly to be appropriate in most cases, necessitating smarter, simpler designs. However, this problem differs from the standard one in that our goal is not only to understand the relationship between the component sequence and the response, but also to screen the components to find the q components that have the largest effect. While only small modifications of the standard order-of-addition models are necessary to accommodate this new component screening problem, new designs are required to keep costs low.

We have proposed three constructions for order-of-addition screening designs and have shown that each guarantees D-optimality for certain run sizes. For the CPS model, Algorithm 1 generates D-optimal designs by leveraging the balance properties of the $\mathbf{F}_{n,m}$ designs generated in Stokes and Xu (2022) for the standard order-of-addition problem. On the other hand, Algorithms 2 and 3 generate D-optimal designs for different choices of m and q under the PWOS model. Algorithm 2 considers the case that m is even and $q = 3$. For this case the algorithm generates highly efficient designs including a D-optimal half-fraction design for both models. Algorithm 3 covers the remaining cases where $m > 4$ and $3 < q < m$. Using the OofA-OA designs developed for the standard order-of-addition problem under the PWO model, this construction generates D-optimal designs with only a small fraction of the number of runs in the full design, with the savings growing larger as m increases. Collectively these three constructions fill an important gap in the order-of-addition literature. In cases where the assumption of a single, fixed q is not met, composite designs can be generated by concatenating the designs from the appropriate construction(s). We see this in the more general version of the job scheduling problem, wherein the sequence that achieves the true minimum across all q values may have a different number of components per sequence than what is assumed in our experiments. A composite design could be beneficial for finding the optimal sequence.

To demonstrate the value of our methods, we have studied several job scheduling problems with job rejection penalties for various values of m

and q. We have found that for simpler problems, training each model on the fractional optimal design from our constructions results in a suitable fit with accurate estimates and strong predictive performance. Furthermore, to showcase the cost-saving potential of our designs, we have considered experiments for job scheduling problems in an active learning framework. The results of this study indicate that the proposed designs tend to result in much faster convergence of the algorithm to the true optimal sequence when compared to a randomly generated initial design, even as m increases. By providing efficient designs for the study of order-of-addition screening experiments, we hope that researchers may soon find many other applications of these results and continue to explore new approaches for modeling this problem. In particular, a two-stage strategy that first screens the components and then studies the ordering effect may be beneficial. For this approach, PWO models that include indicators for whether each component is included in the sequence have shown the potential for capturing the relationship in order-of-addition screening experiments.

Supplementary Materials

The online Supplementary Materials include proofs of the theorems, the algorithm from Stokes and Xu (2022), detailed performance of Algorithm 3, and additional results from the job scheduling problem.

Acknowledgments

The authors thank three reviewers for their helpful comments.

References

Bruni, M., Brusco, L., Ielpa, G. and Beraldi, P. (2019). The risk-averse profitable tour problem. *In Proceedings of the 8th International Conference on Operations Research and Enterprise Systems (ICORES 2019)*,459–466.

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

Garey, M.R., Johnson, D.S., and Sethi, R. (1976). The complexity of jobshop and flowshop scheduling. *Mathematics of Operations Research* **1**(2), 117–129.

Huang, H. (2021). Construction of component orthogonal arrays with any number of components. *Journal of Statistical Planning and Inference* **213**, 72–79.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions, *Journal of Global optimization* **13**, 455–492.

Lin, D. K. J. and Peng, J. (2019). Order-of-addition experiments: A review and some new thoughts. *Quality Engineering* **31**(1), 49–59.

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

Peng, J. Y., Mukerjee, R. and Lin, D. K. J. (2019). Design of order-of-addition experiments. *Biometrika* **106**(3), 683–694.

Schoen, E. D. and Mee, R. W. (2021). Order-of-addition orthogonal arrays to elucidate the sequence effects of treatments. Preprint.

Shabtay, D., Gaspar, N. and Kaspi, M. (2013). A survey on offline scheduling with rejection. *Journal of Scheduling* **16**, 3–28.

Shabtay, D., Gaspar, N. and Yedidsion, L. (2012). A bicriteria approach to scheduling a single machine with job rejection and positional penalties. *Journal of Combinatorial Optimization* **23**, 395–424.

Silvey, S. D. (1980). *Optimal Design*, Chapman and Hall.

Stokes, Z. and Xu, H. (2022). A position-based approach for design and analysis of order-of-addition experiments. *Statistica Sinica*, **32**, doi:10.5705/ss.202020.0178.

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 Nostrand, R. C. (1995). Design of experiments where the order of addition is important. *ASA Proceeding of the Section on Physical and Engineering Sciences*, 155–160. American Statistical Association, Alexandria, Virginia.

Voelkel, J. G. (2019). The design of order-of-addition experiments. *Journal of Quality Technology* **51**(3), 230–241.

Voelkel, J. G. and Gallagher, K. P. (2019). The design and analysis of order-of-addition experiments: An introduction and case study. *Quality Engineering* **31**, 627–638.

Xiao, Q. and Xu, H. (2021). A mapping-based universal Kriging model for order-of-addition experiments in drug combination studies. *Computational Statistics and Data Analysis* **157**, 107155.

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

Zhao, Y., Lin, D. K. J., and Liu, M. Q. (2021a). Designs for order-of-addition experiments. *Journal of Applied Statistics*, **48**(8), 1475–1495.

Zhao, Y., Li, Z., and Zhao, S. (2021b). A new method of finding component orthogonal arrays for order-of-addition experiments. *Metrika*, **84(6)**, 805–824.

Zhong, X., Ou, J., and Wang, G. (2014). Order acceptance and scheduling with machine availability constraints. *European Journal of Operational Research* **232**, 435–441.

Department of Statistics, University of California, Los Angeles, CA 90095, USA

zstokes@ucla.edu

hqxu@stat.ucla.edu