

# Algorithm-Based Optimal and Efficient Exact Experimental Designs for Crossover and Interference Models

SHUAI HAO\*, MIN YANG, AND WEI ZHENG

---

## Abstract

The crossover models and interference models are frequently used in clinical trials, agriculture studies, social studies, etc. While some theoretical optimality results are available, it is still challenging to apply these results in practice. The available theoretical results, due to the complexity of exact optimal designs, typically require some specific combinations of the number of treatments ( $t$ ), periods ( $p$ ), and subjects ( $n$ ). A more flexible method is to build integer programming based on theories in approximate design theory, which can handle general cases of  $(t, p, n)$ . Nonetheless, those results are generally derived for specific models or design problems and new efforts are needed for new problems. These obstacles make the application of the theoretical results rather difficult. Here we propose a new algorithm, a revision of the optimal weight exchange algorithm by [1]. It provides efficient crossover designs quickly under various situations, for different optimality criteria, different parameters of interest, different configurations of  $(t, p, n)$ , as well as arbitrary dropout scenarios. To facilitate the usage of our algorithm, the corresponding R package and an R Shiny app as a more user-friendly interface has been developed.

KEYWORDS AND PHRASES: Crossover, Dropout, Proportional, Interference.

---

## 1. INTRODUCTION

Experimental design plays a fundamental role in scientific research. An optimal design thoughtfully organizes limited experimental resources to ensure reproducible and trustworthy outcomes. This paper focuses on searching for optimal designs for crossover and interference models using a general numerical algorithm proposed by [1].

In a typical experiment with a crossover design, there are  $n$  subjects,  $t$  treatments,  $p$  periods, and  $m(\leq n)$  unique sequences  $s_i$  of length  $p$ . Here,  $n, t, p, m$  are positive integers and we defer the mathematical rigorous definition of the experimental design to a later section. In the experiment, each of the subjects is assigned to a sequence and will receive treatments in a specific order defined in that sequence. Since subjects receive multiple treatments and have data collected throughout all periods, they serve as their own control and typically yield more data compared to other designs with the same number of subjects. Hence, crossover designs are usually more statistically efficient.

However, the limitations are noteworthy as well. First, it is usually unknown if the effects of previous treatments still exist in the following periods. These effects are often referred to as “carryover effects”. It is sensible to use a model that accounts for the carryover effects. There are variants of crossover models, which differ in the way of modeling carryover effects. Section 2 provides details of two of them, and we refer to [2] for an extended list of models. Second, the

competing advantage of crossover designs is contingent on no or minimum dropouts. However in reality, as [3] pointed out “Experience suggests that a dropout rate of between 5% and 10% is not uncommon and, in some areas, can be as high as 25%”. While remedies for missing values due to dropouts have been established, the dropout issue is so common that it has to be prudently considered at the design stage. Related work in this direction includes [3, 4, 5, 6, 7, 8], with a focus on preserving the symmetric structure of the design in the presence of subject dropout. Recently in [9], Kushner’s type of linear equations (see [10]) are developed as necessary and sufficient conditions for a design being universal  $\phi_1$  optimal where  $\phi_1$  is defined as a new surrogate objective function considering dropout. It was later extended to unified results that apply to any configurations of the experiment. Optimal or efficient designs are obtained by either solving linear equations or integer optimization problems modified from equation systems.

The interference model is commonly employed in agricultural studies to mitigate the systematic bias caused by neighbor effects within blocks. A design for interference model consists of  $n$  subjects,  $t$  treatments, and  $m(\leq n)$  distinct blocks  $s_i$  of size  $k$ . Optimal designs can be found in [11, 12, 13], and other sources.

The literature contains abundant theoretical work on optimal exact designs for crossover and interference models. However, despite the elegance of the theorems, they are in general not convenient for use. The optimal or efficient designs are usually derived for a certain combination of  $(n, t, p)$

---

\*Corresponding author.

or  $(n, t, k)$ . In most cases, experimenters need to search for the theory and construct the design themselves. In addition, the relevant theory may not even exist for certain experiment configurations, and this presents a major obstacle to utilizing optimal or efficient designs. Therefore, a general and efficient algorithm easily implemented via a computer program is necessary.

Numerical algorithms have been developed for decades, and it aims at solving the optimal design problem from a general perspective. Existing algorithms are typically modifications of either Fedorov-Wynn algorithm (FWA, [14], [15]), or the multiplicative algorithm (MA, [16]). For more information, refer to [17, 18, 19, 20, 21, 22]. [23] combined multiple existing designs with modifications, named ‘‘cocktail algorithm’’, and achieved dramatic speed improvement. However, all the aforementioned algorithms only focus on D-optimal designs whereas the different objective of experiments requires properly chosen optimality criterion. [1] proposed an optimal weights exchange algorithm (OWEA) for nonlinear models. It updates the support points in the same way as FWA and optimizes weights via the Newton’s method. This is the algorithm we will use.

Nevertheless, the development of an efficient algorithm for crossover designs is faced with two main obstacles. Firstly, it is worth noting that unlike general statistical models where the number of parameters is fixed, the number of parameters in a crossover design varies depending on the number of subjects. This variability extends to the dimension of the information matrix, presenting a significant challenge for algorithm-based optimization. Secondly, constructing optimal crossover designs in the presence of possible subject dropout mechanisms is challenging, considering the variety of crossover design formats. In this paper, we aim to address these challenges.

Specifically, through the Schur complement operation, we derived a novel information matrix that is linear in the design points, with corresponding weights serving as linear coefficients, paving the way for the successful implementation of the OWEA algorithm. Based on [1], we developed a general and efficient algorithm framework for optimal designs of crossover models and later extended it to interference models. Our algorithm is capable of handling any configuration of designs for the two models, including possible subject dropout mechanisms. The designs generated by our approach are highly efficient compared to existing designs in the literature. Additionally, to promote wider usage of our algorithm, we have developed a corresponding R package and R Shiny app, which provides a more user-friendly interface.

The remaining sections of this paper are organized as follows. Section 2 briefly introduces the necessary notations, statistical models and the derivations of the information matrices to fit into the framework of the algorithm. Section 3 gives details of the algorithm and the developed R-package. Numerical results are presented in section 4, and a short discussion is in section 5.

## 2. THE FRAMEWORK AND NOTATIONS

In this section, we will introduce the general framework starting with basic concepts of optimal designs followed by demonstrations of crossover models and interference models as well as their information matrix. We will show how the information matrices, utilizing Schur complement, fit into the algorithmic framework in [1].

### 2.1 Optimal Designs and Objective Functions

Despite the various types of designs, generally, an experimental design with  $n$  runs can be written as

$$d = \{(s_i, n_i) | s_i \in \chi, \sum_{i=1}^m n_i = n, n_i > 0\}, \quad (2.1)$$

where  $(s_i, n_i)$  is the pair of the design point and its associated repetition. The  $s_i$  is a vector of length  $p (\geq 1)$ . The *design space*, which is a collection of the design points, is represented by  $\chi$ . The  $m$  is the number of distinct design points. If all of the repetitions  $(n_1, \dots, n_m)$  are required to be positive integers, then the design (2.1) is called *exact design* and  $m \leq n$ .

To derive an optimal design is to find a set of pairs  $(s_i, n_i)$  that optimize an objective function. Due to the discrete nature of the exact design, integer solutions of  $n_i$  are generally intractable. In that case, one may drop the integer requirement and search for an approximate design:

$$\xi = \{(s_i, w_i) | s_i \in \chi, \sum_{i=1}^m w_i = 1, w_i \in (0, 1]\}. \quad (2.2)$$

where  $w_i$  is the weight associated to design point  $s_i$  and all weights sum up to unity.

In after-experiment data analysis, usually a differentiable function of  $\theta$ , say  $g(\theta)$ , is of interest. Notice that  $g$  must be an estimable function. In the rest of this paper, we always have this assumption. The choice of  $g$  grants the flexibility on parameter of interest. For example, when we target on all parameters,  $g$  is an identity function, i.e.  $g(\theta) = \theta$ ; and when comparison is the goal, one of many options of  $g$  is  $g(\theta) = (\theta_1 - \theta_\nu, \theta_2 - \theta_\nu, \dots, \theta_{\nu-1} - \theta_\nu)$ , where  $\nu$  is the length of  $\theta$ . Denote by  $\hat{\theta}$  the maximum likelihood estimator (MLE) of  $\theta$ , then it is well known that  $g(\hat{\theta})$  is also the MLE of  $g(\theta)$ . Under mild assumptions, the Delta method yields the asymptotic covariance matrix of  $g(\hat{\theta})$ . With  $\mathbf{I}_\xi(\theta)$  being the information matrix of  $\theta$  under design  $\xi$ ,  $C_\xi(g)$  is the asymptotic variance-covariance matrix of  $g(\hat{\theta})$  obtained using the Delta method under mild conditions,

$$C_\xi(g) = \frac{\partial g}{\partial \theta'} \mathbf{I}_\xi^{-1}(\theta) \left( \frac{\partial g}{\partial \theta'} \right)'. \quad (2.3)$$

With the notations of (2.1) and (2.2), we follow the definition in [24] of the objective function,

$$\Phi_r(C_\xi(g)) = \left[ \frac{1}{v} \text{Tr}(C_\xi(g))^r \right]^{1/r}, \quad 0 \leq r < \infty, \quad (2.4)$$

where  $r$  is a non-negative real number,  $v$  is the dimension of  $g$ ,  $g$  is a differentiable function of parameter vector  $\theta$ , and  $Tr(C_\xi(g))$  is the trace of the matrix  $C_\xi(g)$ .

Note that  $\Phi_r(C_\xi(g))$  is equivalent to many prevailing optimality criteria on various values of  $r$ . For example,  $r = 0$ ,  $\Phi_r(C_\xi(g))$  is  $\lim_{r \downarrow 0} [\frac{1}{v} Tr(C_\xi(g))^r]^{1/r} = |C_\xi(g)|^{1/v}$ , which is D-optimality; when  $r = 1$ , we have  $\Phi_r(C_\xi(g)) = Tr(C_\xi(g))/v$ , and it is A-optimality.

For the purpose of verifying a design is optimal, the following theorem, adapted from Theorem 1 of [1], provides Kiefer's type general equivalence theorem (GET) which serves as a sufficient and necessary condition.

**Theorem 1.** *Let  $g(\theta)$  be an estimable function of  $\theta$ . Suppose an arbitrary design  $\xi$  has an information matrix  $\mathbf{I}_\xi$ .  $\xi$  is  $\Phi_r$  optimal for  $g(\theta)$  if and only if the directional derivative of  $\Phi_r$ , denoted by  $d_r(s, \xi)$  satisfies*

$$d_r(s, \xi) \leq 0, \quad (2.5)$$

for any  $s \in \chi$ , with equality holds if  $s$  belongs to the support of  $\xi$ .

In addition, the  $d_r(s, \xi)$  can be calculated by: when  $r = 0$ ,

$$d_r(s, \xi) = Tr[(C_\xi(g))^{-1} (\frac{\partial g}{\partial \theta}) \mathbf{I}_\xi^- (\mathbf{I}_s - \mathbf{I}_\xi) \mathbf{I}_\xi^- (\frac{\partial g}{\partial \theta})']. \quad (2.6)$$

when  $r > 1$ ,

$$d_r(s, \xi) = (\frac{1}{v})^{1/r} Tr[(C_\xi(g))^{1-r} \times Tr[(C_\xi(g))^{r-1} (\frac{\partial g}{\partial \theta}) \mathbf{I}_\xi^- (\mathbf{I}_s - \mathbf{I}_\xi) \mathbf{I}_\xi^- (\frac{\partial g}{\partial \theta})']]. \quad (2.7)$$

## 2.2 Models and Information Matrix

### 2.2.1 Crossover Model with Carryover Effect

Suppose a crossover design  $d$  is conducted with  $n$  subjects,  $t$  treatments and  $p$  periods, the observation from subject  $i$  at period  $j$  is typically modeled by

$$y_{ij} = \mu + \zeta_i + \pi_j + \tau_{d(i,j)} + \gamma_{d(i,j-1)} + \epsilon_{ij}; \quad (2.8)$$

$$i = 1, \dots, n; \quad j = 1, \dots, p,$$

where  $\mu$  is the grand mean,  $\zeta_i$  is the effect from  $i$ th subject,  $\pi_j$  stands for the effects from  $j$ th period,  $d(i, j)$  denotes the treatment assignment of  $j$ th period for  $i$ th subject from design  $d$ ,  $\tau_{d(i,j)}$  is the treatment effect from  $d(i, j)$ , and  $\gamma_{d(i,j-1)}$  is the carryover effect due to treatment  $d(i, j-1)$  where  $\gamma_{d(i,0)}$  is set to 0 by convention.  $\epsilon_{ij}$  is the error term.

If the collection of observations of the design  $d$  is organized in a vector  $Y_d = (y_{11}, y_{12}, \dots, y_{1p}, y_{21}, \dots, y_{np})'$ , then model (2.8) can be written in matrices,

$$Y_d = \mathbf{1}_{np} \mu + U \zeta + Z \pi + T \tau + R \gamma + \epsilon, \quad (2.9)$$

where  $\epsilon$  is a vector of independent and identically distributed errors with a mean vector  $\mathbf{0}$  and a variance-covariance matrix  $\sigma^2 I_{np}$ , which can be relaxed to a form of Kronecker's product,  $I_n \otimes \Sigma$ , in most cases when the variance-covariance matrix of  $(\epsilon_{i1}, \dots, \epsilon_{ip})$  is  $\Sigma$  for  $i = 1, \dots, n$ . Besides, we also have  $\pi = (\pi_1, \dots, \pi_p)'$ ,  $\zeta = (\zeta_1, \dots, \zeta_n)'$ ,  $\tau = (\tau_1, \dots, \tau_t)'$ ,  $\gamma = (\gamma_1, \dots, \gamma_t)'$ ,  $U = I_n \otimes \mathbf{1}_p = (U'_1, \dots, U'_n)'$ ,  $Z = \mathbf{1}_n \otimes I_p = (Z'_1, \dots, Z'_n)'$ ,  $T = (T'_1, \dots, T'_n)'$  and  $R = (R'_1, \dots, R'_n)'$ . Here  $U_i$ 's are  $p \times n$  incidence matrices for subjects.  $Z_i$  is  $p \times p$  subject-to-period incidence matrix,  $T_i$  and  $R_i$  are  $p \times t$  period-to-treatment incidence matrices for the  $i$ th subject. The structure of matrices  $Z_i$ ,  $T_i$ , and  $R_i$  depend on the design  $d$ .  $I_n$  is an identity matrix of dimension  $n$  and  $\mathbf{1}_p$  stands for a  $p \times 1$  vector of 1's. Then the information matrix for full parameter vector  $\Theta = (\zeta', \pi', \tau', \gamma)'$  is

$$\mathbf{I}(\Theta) = (U \quad Z \quad T \quad R)' (U \quad Z \quad T \quad R) \quad (2.10)$$

$$= \sum_{i=1}^n (U_i \quad Z_i \quad T_i \quad R_i)' (U_i \quad Z_i \quad T_i \quad R_i) \quad (2.11)$$

$$= \sum_{i=1}^n \mathbf{I}_i = \sum_{i=1}^m n_i \mathbf{I}_i,$$

where the positive integer  $m (\leq n)$  represents the number of unique design point (sequence of treatments) of the design, and  $m = n$  holds when there is only one subject assigned to each of the unique design points.  $\mathbf{I}_i$  is the information matrix for the  $i$ th unique design point, and  $n_i$  is the number of subjects assigned to that design point or alternatively the "repetition" of a design point. Additionally,  $\sum_i n_i = n$ . For the remaining of this paper, unless otherwise specified, the term "design point" refers to "unique design point" in an experimental design.

An optimal exact design is then to choose a collection of sequences so that it optimizes an objective function related to (2.10). In this paper, we obtain optimal approximate design by a numerical algorithm and carefully round it to an exact design which is either efficient or optimal. However two issues regarding information matrix has to be settled prior to the implementation of the algorithm. First, the direct treatment is the target to estimate, and yet its information matrix is not additive with respect to design points. As a result, it does not fit for the prerequisite of the algorithm. On the other hand, although information matrix for full parameters is additive, its dimension is changing all the time because of the inclusion of subject effects  $\zeta$  as well as the iterative nature of numerical algorithm. Optimizing on an information matrix of varying dimensions is problematic. Therefore, to fit the crossover design problem to our framework, we exclude the subject effect from (2.10) through the Schur complement operation. Excluding them and the general mean (noticing that the unity vector  $\mathbf{1}_{np}$  belongs to the column space of  $U$ ) and utilizing the Schur complement of a matrix, we reach to the information matrix of the marginal

parameter vector  $\theta = (\pi', \tau', \gamma)'$  as

$$\begin{aligned} \mathbf{I}(\theta) &= (Z \ T \ R)' pr^\perp(U) (Z \ T \ R) \\ &= (Z \ T \ R)' [I_{np} - \frac{1}{p} I_n \otimes 1_p 1_p'] (Z \ T \ R) \\ &= \sum_{i=1}^n (Z_i \ T_i \ R_i)' [I_p - \frac{1}{p} 1_p 1_p'] (Z_i \ T_i \ R_i) \\ &= \sum_{i=1}^n \mathbf{I}_i = \sum_{i=1}^m n_i \mathbf{I}_i = n \sum_{i=1}^m w_i \mathbf{I}_i, \end{aligned} \tag{2.12}$$

where,  $n_i = nw_i$ , for  $i = 1, 2, \dots, m$ , and  $w_i$  is called the “weight” associated to a design point.  $pr^\perp(\cdot)$  is an orthogonal projection operator. Notice that  $\mathbf{I}_i$  is the information matrix associated with the  $i$ th design point. For any matrix  $X$ ,  $pr^\perp(X) = I - X(X'X)^{-1}X'$ . With such derivation, the information matrix is linear in the design points with corresponding weights as linear coefficients.

### 2.2.2 Crossover Model with Dropout

To account for the dropouts, following [3, 9], define the dropout mechanism of a subject as  $\ell = (\ell_1, \dots, \ell_p)$ , where  $\ell_i$  is the probability that the longest time of stay is  $i$ , and  $\sum_{i=1}^p \ell_i = 1$  and assume

1. Subject’s dropout is independent of the choice of design as well as the outcome of the design.
2. Once a subject is dropped, the chance of returning is zero.
3. Subject’s dropout mechanisms are i.i.d.

Given a realization of  $\ell$ , the information matrix (2.12) becomes

$$\begin{aligned} \mathbf{I}(\theta, \ell) &= \sum_{i=1}^n (Z_i \ T_i \ R_i)' [M_i - \\ &\quad M_i 1_p (1_p' M_i 1_p)^{-1} 1_p' M_i] (Z_i, T_i, R_i) \\ &= \sum_{i=1}^n \mathbf{I}_i(\ell) = \sum_{i=1}^m n_i \mathbf{I}_i(\ell) = n \sum_{i=1}^m w_i \mathbf{I}_i(\ell), \end{aligned} \tag{2.13}$$

where  $M_i$  is an indicator matrix depending on the dropout mechanism  $\ell$ ,

$$M_i(\ell) = \begin{pmatrix} I_{(a_i \times a_i)} & O_1 \\ O_1 & O \end{pmatrix}_{p \times p}, \tag{2.14}$$

$I_{(a_i \times a_i)}$  is the  $a_i \times a_i$  identity matrix,  $a_i$  is the longest period of stay for subject  $i$ , and  $O$  and  $O_1$  are zero matrices with proper dimensions.

The information matrix is random due to the modeling of the dropout mechanism. The intuitively reasonable objective function,  $E\Phi_p(C_\xi(g))$  (defined in (2.4)), is not easy to deal with, and  $\Phi_p(E(C_\xi(g)))$  is chosen as a feasible replacement. Here the operator  $E$  means expectation with respect to the dropout mechanism. Suppose we obtain an optimal approximate design  $\xi^*$  by minimizing  $\Phi_p(E(C_\xi(g)))$ ,

and round it to exact design  $\xi_1$ . The efficiency of  $\xi_1$  should be defined as  $E\Phi_p(C_{\xi_{opt}}(g))/E\Phi_p(C_{\xi_1}(g))$ , where  $\xi_{opt} = \underset{\xi}{\operatorname{argmin}} E\Phi_p(C_\xi(g))$ . However, the efficiency is not accessible because  $\xi_{opt}$  is unknown. Nevertheless, one is still able to obtain the lower bound.

**Lemma 1** (Zheng (2013a)). *For crossover designs under subject dropout, define the following approximate designs*

$$\xi_{opt} = \underset{\xi}{\operatorname{argmin}} E\Phi_p(C_\xi(g)), \tag{2.15}$$

$$\xi^* = \underset{\xi}{\operatorname{argmin}} \Phi_p(E(C_\xi(g))). \tag{2.16}$$

For an arbitrary design  $\xi_1$ ,

$$\frac{E\Phi_p(C_{\xi_{opt}}(g))}{E\Phi_p(C_{\xi_1}(g))} \geq \frac{\Phi_p(E(C_{\xi_{opt}}(g)))}{E\Phi_p(C_{\xi_1}(g))} \geq \frac{\Phi_p(E(C_{\xi^*}(g)))}{E\Phi_p(C_{\xi_1}(g))},$$

where the first ‘ $\geq$ ’ is due to the convexity of the function  $\Phi_p$  and Jensen’s inequality, and the second ‘ $\geq$ ’ is because of (2.15). Note that the problem in (2.16) fits the framework of (2.12) and hence could be solved by our algorithm. Therefore, the rightmost term,  $\frac{\Phi_p(E(C_{\xi^*}(g)))}{E\Phi_p(C_{\xi_1}(g))}$ , becomes evaluable for any given arbitrary design  $\xi_1$  and dropout mechanism and will serve as the lower bound efficiency. Furthermore, [9] proposed to convert the results of (2.16) to an exact design and gauge its performance by the lower bound of efficiency. We are using the same approach except that our algorithm to derive the exact design is much faster.

### 2.2.3 Crossover Model with Proportional Carryover Effects

Another variant of the classical crossover design model is the proportional model. For the design  $d$  with  $p$  periods and  $t$  treatments, individual outcomes are modeled as

$$y_{ij} = \mu + \zeta_i + \pi_j + \tau_{d(i,j)} + \lambda\tau_{d(i,j-1)} + \epsilon_{ij}; \tag{2.17}$$

$i = 1, \dots, n; \quad j = 1, \dots, p,$

where the notations are exactly the same as in (2.8), except that the additional real-valued  $\lambda$ , which is the proportion that carryover effects accounting for direct effects. The proportional model can also be written in matrices,

$$Y_d = 1_{np}\mu + U\zeta + Z\pi + T\tau + \lambda R\tau + \epsilon, \tag{2.18}$$

where  $\epsilon$  is a vector of independent errors with mean  $\mathbf{0}$  and variance covariance matrix  $I_n \otimes \Sigma$ ,  $\Sigma$  is an  $p \times p$  positive definite matrix. Utilizing Schur complement operations, the information matrix for the parameter vector  $\theta = (\pi', \tau', \lambda)'$  is

$$\begin{aligned} \mathbf{I}(\theta, \lambda, \tau) &= \sum_{i=1}^n (Z_i, T_i + \lambda R_i, R_i \tau)' [\Sigma^{-1} - \\ &\quad \Sigma^{-1} 1_p [1_p' \Sigma^{-1} 1_p]^{-1} 1_p' \Sigma^{-1}] (Z_i, T_i + \lambda R_i, R_i \tau) \\ &= \sum_{i=1}^n \mathbf{I}_i = \sum_{i=1}^m n_i \mathbf{I}_i = n \sum_{i=1}^m w_i \mathbf{I}_i. \end{aligned} \tag{2.19}$$

Therefore, the design problem for the proportional model fits into the framework of (2.12). Note that the proportional model is nonlinear due to the term  $\lambda\tau$ , and its information matrix depends on unknown parameters  $\tau$  and  $\lambda$ .

### 2.3 Interference Model

The individual observations of the interference model with  $t$  treatments,  $n$  blocks of size  $k$  are written as

$$y_{ij} = \mu + \gamma_i + \tau_{d(i,j)} + \lambda_{d(i,j-1)} + \rho_{d(i,j+1)} + \epsilon_{ij} \quad (2.20)$$

$$i = 1, \dots, n; \quad j = 1, \dots, k,$$

where  $y_{ij}$  denotes the response from  $j$ th plot of  $i$ th block,  $\mu$  is general mean,  $d(i, j)$  stands for the treatment assignment of  $i$ th block and  $j$ th plot according to design  $d$ , and  $\tau_{d(i,j)}$ ,  $\lambda_{d(i,j-1)}$ , and  $\rho_{d(i,j+1)}$  are treatment effects from treatment itself, its left plot and right plot. By convention, we set  $\lambda_{d(i,0)} = \rho_{d(i,p+1)} = 0$ . Again, the responses can be gathered in a vector and modeled in terms of matrices,

$$Y_d = 1_{nk}\mu + U\gamma + T_d\tau + L_d\lambda + R_d\rho + \epsilon, \quad (2.21)$$

where  $Y_d = (y_{11}, \dots, y_{1k}, y_{21}, \dots, y_{np})'$ ,  $\gamma = (\gamma_1, \dots, \gamma_n)'$ ,  $\tau = (\tau_1, \dots, \tau_t)'$ ,  $\lambda = (\lambda_1, \dots, \lambda_t)'$ ,  $\rho = (\rho_1, \dots, \rho_t)'$ ,  $U = I_n \otimes 1_k$ ,  $T_d = (T'_1, \dots, T'_n)'$ ,  $L_d = (L'_1, \dots, L'_n)'$ ,  $R_d = (R'_1, \dots, R'_n)'$ . Here  $U_i$ 's are incidence matrices for block,  $T_i$  and  $L_i$  and  $R_i$  are  $k \times t$  plot-to-treatment incidence matrices for  $i$ th block that depends on the design  $d$ . The error vector  $\epsilon$  is assumed to follow  $N(0, I_n \otimes \Sigma)$ , where  $\Sigma$  is an arbitrary positive definite matrix of order  $k$ . Let  $\theta = (\tau', \lambda', \rho)'$  be the vector of parameters of interest. Note that block effects are excluded due to similar reasons to that of the subject effect for the crossover model. The information matrix for  $\theta$ , after applying Schur complement, is

$$\mathbf{I}(\theta) = \sum_{i=1}^n (T_i, L_i, R_i)' [\Sigma^{-1} - \Sigma^{-1} 1_k (1'_k \Sigma^{-1} 1_k)^{-1} 1'_k \Sigma^{-1}] (T_i, L_i, R_i)$$

$$= \sum_{i=1}^n \mathbf{I}_i = \sum_{i=1}^m n_i \mathbf{I}_i = n \sum_{i=1}^m w_i \mathbf{I}_i. \quad (2.22)$$

At this point, we have fitted the design problem for the crossover model with dropout, the proportional model and the interference model into the framework of (2.12), to which our algorithm would apply. In the next section, we shall introduce the algorithm and the developed R-package.

## 3. THE ALGORITHM AND R-PACKAGE

Numerical algorithms are powerful and convenient tools for finding optimal designs. [1] proposed an optimal weights exchange algorithm (OWEA) for nonlinear models. It updates the support points in the same way as Fedorov-Wynn algorithm and optimizes weights via the Newton's method.

This is the algorithm we are going to use. The procedures of OWEA's implementation is briefly introduced as follows.

### Phase I: Finding optimal approximate designs

Start with initial support  $S^{(1)}$  with  $\nu + 1$  randomly picked sequences and equal weights  $w^{(1)}$ , at iteration  $t$ ,

1. Input support  $S^{(t)}$ ,  $w^{(t)}$ , and update weights using the Newton's method. Note that support points with zero weights will be deleted in optimizing process.

#### Newton's Method:

**Input:** Start with the initial input by setting  $S_1^{(t)} = S^{(t)}$ , and  $w_1^{(t)} = w^{(t)}$ . After  $j-1$  iterations, with the  $S_j^{(t)}$  and  $w_j^{(t)}$ , updating the weight by the following steps:

- (a) update the weights by the equation  $w_{j+1}^{(t)} = w_j^{(t)} - \alpha \left( \frac{\partial^2 \Phi}{\partial w \partial w'} \right)^{-1} \frac{\partial \Phi}{\partial w} \Big|_{w_j^{(t)}}$ .
- (b) If  $w_{j+1}^{(t)}$  has negative component, go to step (c), otherwise proceed to step (d).
- (c) Set  $\alpha$  to  $\alpha/2$  and go back to step (a). If  $\alpha < \epsilon_\alpha$ , remove the point with smallest weight and go back to step (a).
- (d) Check if  $\frac{\partial \Phi}{\partial w} \Big|_{w_{j+1}^{(t)}} < \epsilon_0$ , if true,  $\tilde{w}^{(t)} = w_{j+1}^{(t)}$  is the optimal weights otherwise go to next iteration.

**Output:** Support  $\tilde{S}^{(t)}$  and optimal weights  $\tilde{w}^{(t)}$ , where  $\tilde{S}^{(t)} = S^{(t)}$  if no points are removed.

2. Derive  $s_t^* = \underset{s \in \mathcal{X}}{\operatorname{argmax}} d_r(s, \xi^{(t)})$ , where  $\xi^{(t)} = \{(s_i, w_i) | s_i \in \tilde{S}^{(t)}, w_i \in \tilde{w}^{(t)}, \sum_i w_i = 1\}$
3. Check  $d_r(s_t^*, \xi^{(t)}) < \epsilon_d$ , where  $\epsilon_d$  is a pre-selected small positive value. If true,  $\xi^{(t)}$  is the desired design. Otherwise, let  $S^{(t+1)} = \tilde{S}^{(t)} \cup \{s_t^*\}$ ,  $w^{(t+1)} = \tilde{w}^{(t)} \cup \{0\}$ , and go to step 1.

### Phase II: Rounding

Suppose optimal design from Phase I is  $\xi^* = \{(s_i, w_i^*) | s_i \in S^*, w_i \in (0, 1), \sum_i w_i = 1\}$ , for a given number of total runs, say  $N$ ,

1. Compute  $Nw_i$  for  $i = 1, \dots$ , and round them to their nearest integers. This results in a tentative exact design  $d = \{(s_i, n_i) | s_i \in S^*, \sum_i n_i = N\}$
2. For every  $s_j \in \Omega/S^*$ ,
  - (a) For every unique  $s_i \in S^*$ , subtract  $n_i$  by 1 and append  $s_j$  to the design with repetition 1, this results in a new design, say  $d^{(-i,j)}$ .
  - (b) Calculate two values  $V1 = \Phi(d)$  and  $V2 = \Phi(d^{(-i,j)})$ , if  $V1 > V2$ , keep  $d^{(-i,j)}$  as the current tentative design, otherwise, keep the tentative design unchanged.
  - (c) Repeat these two steps for  $i = 1, \dots, m$ ;

Table 1.  $t = p = 4, n = 16, a = (0,0,1/2,1/2)$ .

Optimality	Rel-Eff (to Gurobi)	Rel-Eff (rounding)	Efficiency (Lower Bound)	Rel-Eff (to $d_2$ in [25])	Time (sec)	Gurobi Time (sec)
A	0.9994	0.9917	0.9366	0.9844	2.18	20.84
D	0.9979	0.9757	0.9519	0.9538	2.63	20.84

3. After traversing all possible  $j$ 's, output the current design as efficient design  $d^*$ .

Note that tuning parameters  $\epsilon_d, \epsilon_\alpha$  and  $\epsilon_0$  may affect the time for convergence. In all examples tested in this paper, the combination of  $\epsilon_d = 10^{-15}, \epsilon_\alpha = 10^{-6}$  and  $\epsilon_0 = 10^{-10}$  work well and optimal or highly efficient designs can be found within a relatively short period of time. Besides, it is known that the Newton's method is sensitive to the choice of initial points and sometimes has overshooting issue. However, based on our experience, efficient designs are always found.

We have developed an R Package called OWEA to apply the algorithm to the three models as mentioned in the earlier section, namely (i) crossover design accounting for subject dropouts; (ii) proportional model; (iii) interference model. It can be readily applicable to all these three optimal design problems. To implement it, a user only needs to input very basic information such as numbers of periods, treatments, and subject, and of course the dropout probability for the model in (i). The detailed implementation is available at <https://cran.r-project.org/package=OWEA>.

### 4. EXAMPLES

In this section, utilizing the information matrices derived in section 2 and the OWEA algorithm in Section 3, exact designs are presented for three scenarios: crossover model with dropouts, crossover model with proportional carryover effects, and interference models. For convenience, we call the design obtained from OWEA algorithm the *OWEA design*, optimal design available from the literature the *literature design*. For example, the A- optimal design obtained from OWEA is "A-optimal OWEA design". Moreover, for all the three scenarios, [9, 13, 25] developed linear equation systems as necessary and sufficient conditions for universally optimal designs and suggested searching for optimal designs using integer optimizer, like [26]. More specifically, the conditions are in the form of linear equations regarding the replication numbers of each treatment sequence. Since there is no guarantee of integer solutions, we transform the problem into a quadratic programming which minimizes the total square of loss for any equation that does not hold. In this section, we call designs derived such way as the *Gurobi design*.

Relative efficiency, shown as "Rel-Eff (to xx)", is provided when the intention is to compare the OWEA designs to existing designs. In addition, the relative efficiency due to rounding from the approximate design to the exact design, shown as "Rel-Eff (rounding)", is also presented.

All the algorithms and examples are programmed in R (R Core Team, 2022) [27] and are executed on an Apple Desktop with 8GB RAM. Computing times are recorded and provided.

#### 4.1 Crossover with Subject Dropout

**Example 1.**  $(t, p) = (4, 4), n = 16, \ell = (0, 0, 1/2, 1/2)$ . Exact designs and their performance under A- and D- optimality are summarized in Table 1. It is evident that there is only a tiny loss in efficiency when rounding from approximate design to exact design, though the lower bound efficiencies are only 0.9366 and 0.9519 for the A- and D- criteria. Compared to  $d_2$  from [9] which is proved to be highly efficient, the relative efficiency of the OWEA designs are 0.9844 and 0.9538 respectively. They are not very high but still satisfactory. The design obtained utilizing the integer programming from [9] is also available and it is almost identical to the OWEA designs in efficiency, but the OWEA framework only takes a small fraction of computing time.

$$A - opt = \begin{matrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\ 2 & 3 & 4 & 4 & 1 & 1 & 3 & 4 & 1 & 2 & 2 & 4 & 1 & 2 & 2 & 3 \\ 3 & 4 & 3 & 3 & 3 & 4 & 1 & 1 & 2 & 4 & 4 & 2 & 2 & 1 & 3 & 1 \\ 4 & 2 & 2 & 2 & 4 & 3 & 4 & 3 & 4 & 1 & 1 & 1 & 3 & 3 & 1 & 2 \end{matrix} \quad (4.1)$$

$$D - opt = \begin{matrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\ 2 & 3 & 4 & 4 & 1 & 1 & 3 & 4 & 1 & 2 & 2 & 4 & 1 & 2 & 2 & 3 \\ 3 & 4 & 3 & 3 & 3 & 4 & 1 & 1 & 2 & 4 & 4 & 2 & 2 & 1 & 3 & 1 \\ 4 & 2 & 2 & 2 & 4 & 3 & 4 & 3 & 4 & 1 & 1 & 1 & 3 & 3 & 1 & 2 \end{matrix} \quad (4.2)$$

$$d_2 = \begin{matrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\ 2 & 2 & 3 & 4 & 1 & 3 & 4 & 4 & 1 & 1 & 2 & 4 & 1 & 2 & 3 & 3 \\ 3 & 3 & 4 & 2 & 4 & 1 & 1 & 3 & 2 & 4 & 4 & 1 & 3 & 1 & 2 & 2 \\ 4 & 4 & 2 & 2 & 4 & 1 & 3 & 3 & 2 & 2 & 4 & 1 & 3 & 3 & 1 & 1 \end{matrix} \quad (4.3)$$

$$\text{Gurobi Design} = \begin{matrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\ 2 & 2 & 3 & 4 & 1 & 3 & 4 & 4 & 1 & 2 & 4 & 4 & 1 & 2 & 3 & 3 \\ 3 & 3 & 2 & 2 & 4 & 1 & 1 & 3 & 2 & 4 & 1 & 2 & 3 & 1 & 1 & 2 \\ 4 & 4 & 4 & 2 & 4 & 4 & 3 & 3 & 2 & 4 & 1 & 1 & 3 & 3 & 1 & 1 \end{matrix} \quad (4.4)$$

**Example 2.**  $(t, p) = (4, 4), n = 19, \ell = (0, 0, 1/2, 1/2)$ . This is a case where  $n$  is not a multiple of either  $t$  or  $p$ , and the optimal designs in literature are not available. Exact OWEA designs and the Gurobi designs are summarized in Table 2. Still, the lower bound efficiency is around 0.95, but the efficiency loss due to rounding is less than 0.015. In addition, the OWEA designs outperform the Gurobi designs with improved efficiency of 1.0261 and 1.0785 under

Table 2.  $t = p = 4, n = 19, a = (0,0,1/2,1/2)$ .

Optimality	Rel-Eff (to Gurobi)	Rel-Eff (Rounding)	Efficiency (Lower Bound)	Time (sec)	Gurobi Time (sec)
A	1.0261	0.9944	0.9489	2.28	100.41
D	1.0785	0.9855	0.9576	2.71	100.41

A- and D- optimal criteria but with only no more than 3% of computing time.

$$A - opt = \begin{matrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 \\ 3 & 3 & 4 & 4 & 4 & 1 & 1 & 3 & 3 & 4 & 1 & 2 & 2 & 4 & 1 & 1 & 2 & 2 & 3 \\ 2 & 4 & 3 & 3 & 3 & 3 & 4 & 1 & 4 & 1 & 2 & 4 & 4 & 2 & 2 & 2 & 1 & 3 & 1 \\ 4 & 2 & 2 & 2 & 2 & 4 & 3 & 4 & 1 & 3 & 4 & 1 & 1 & 1 & 3 & 3 & 3 & 1 & 2 \end{matrix} \quad (4.5)$$

$$D - opt = \begin{matrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 \\ 2 & 3 & 3 & 4 & 4 & 1 & 1 & 3 & 3 & 4 & 1 & 2 & 2 & 4 & 4 & 1 & 1 & 2 & 3 \\ 3 & 2 & 4 & 3 & 3 & 3 & 4 & 1 & 1 & 1 & 4 & 4 & 4 & 2 & 2 & 2 & 2 & 1 & 1 \\ 4 & 4 & 2 & 2 & 2 & 4 & 3 & 4 & 4 & 3 & 2 & 1 & 1 & 1 & 1 & 3 & 3 & 3 & 2 \end{matrix} \quad (4.6)$$

$$\text{Gurobi Design} = \begin{matrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 3 & 4 & 4 & 4 & 1 & 1 & 3 & 3 & 4 & 4 & 1 & 2 & 2 & 4 & 1 & 1 & 3 & 2 & 2 \\ 4 & 4 & 2 & 2 & 3 & 3 & 1 & 1 & 1 & 3 & 2 & 1 & 4 & 1 & 2 & 2 & 2 & 1 & 1 \\ 3 & 4 & 3 & 3 & 3 & 4 & 1 & 4 & 3 & 3 & 2 & 4 & 4 & 1 & 3 & 3 & 2 & 3 & 3 \end{matrix} \quad (4.7)$$

A & D optimal	
Sequence	Repetition
QRST	9
QRTS	12
QSTR	15
QTRS	26
QTSR	1
RQTS	8
RSTQ	28
RTQS	5
RTSQ	22
SQRT	24
SRQT	29
STRQ	9
TQSR	30
TRQS	12
TSQR	16
TSRQ	4

**Example 3.**  $(t, p) = (4, 4), n = 250$ . This is a real world application based on [28]. The *Best African American Response to Asthma Drugs (BARD) Trial*, contained two four-treatment, four-period, four-sequence crossover trials, one involving 294 adolescents and adults, and the other one involving 280 children. For the purpose of illustration, we focus on the BARD trial for the children. The four treatment regimens were represented by letters Q, R, S, and T. A total of 280 participants were evenly randomized to one of the four sequences, QRST, RTQS, SQTR and TSRQ. The study design was uniform and balanced. Out of the 280 children: 30 dropped out before period 1, 16 dropped out after period 1, 17 dropped out after period 2, 19 dropped out after period 3, and the remaining 198 completed the study. Since the subjects who drop out before period 1 provides no information at all, for the illustration of our method, we shall study the optimal design of crossover design with  $(280 - 30 =)250$  subjects, 4 periods, 4 treatments, and the adjusted dropout mechanism as  $\ell = (16, 17, 19, 198)/250 = (0.064, 0.068, 0.076, 0.792)$ . The resultant OWEA designs as well as their performances are available in Table 3.

The OWEA exact design under A- and D- optimality happened to be exactly the same. Their relative efficiency due to rounding are almost 1, with lower bound efficiencies being all above 0.99. The relative efficiencies as compared to designs in [28] are 1.0023 for A-criterion and 1.0063 for D-criterion. This means that the OWEA framework improves the design efficiency after incorporating the dropout mechanism. Finally, the computing time is around 2.20 seconds under both criteria.

## 4.2 Crossover Model with Proportional Carryover Effects

**Example 4.**  $(t, p) = (3, 3), n = 36$ . The proportional model is nonlinear. All OWEA designs in this paper are locally optimal. For this particular example, we assume all direct treatment effects to be 2 and  $\lambda = 0.2$ , and the covariance structure is taken to be the identity matrix. According to Table 4, the OWEA designs have identical efficiency under A- and D- optimal criteria compared to the optimal designs in the literature. In fact, the OWEA design is equivalent to the symmetric block  $\langle 123 \rangle$  defined in [25] and it is shown to have unity efficiency for A- D- T- criteria and 0.9931 under E-optimality. Besides, the Gurobi design is also calculated, and comparably it is less efficient. Trivially, the difference between computing times is negligible as both algorithms take less than 0.1 seconds.

$$A - opt = \begin{matrix} & 1 & 2 & 3 \\ 2 & 3 & 1 & \\ & 3 & 1 & 2 \end{matrix} \times 12 \quad (4.8)$$

$$D - opt = \begin{matrix} & 1 & 2 & 3 \\ 2 & 3 & 1 & \\ & 3 & 1 & 2 \end{matrix} \times 12 \quad (4.9)$$

$$\text{Gurobi Design} = \begin{cases} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 3 & 3 & 3 & 3 & 3 & 2 & 2 & 2 & 2 & 2 & 3 & 1 & 3 & 3 & 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 3 & 1 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 2 \end{cases} \quad (4.10)$$

Table 3.  $t = p = 4, n = 250, a = (0.064, 0.068, 0.076, 0.792)$ .

Optimality	Rel-Eff (rounding)	Efficiency (Lower Bound)	Rel-Eff (to [28] with $n = 250$ )	Time (sec)
A	0.9999	0.9977	1.0023	2.24
D	0.9999	0.9916	1.0063	2.15

Table 4.  $t = p = 3, n = 36$ .

Optimality	Rel-Eff (to Gurobi)	Rel-Eff (to $d_2$ in [25])	Time (sec)	Gurobi Time (sec)
A	1.0390	1	0.01	0.06
D	1.0794	1	0.01	0.06

Table 5.  $t = 4, p = 3, n = 12$ .

Optimality	Rel-Eff (to Gurobi)	Rel-Eff (to $d_1$ in [29])	Time (sec)	Gurobi Time (sec)
A	1.009	0.9980	0.17	0.03
D	1.027	0.9997	0.22	0.03

Table 6.  $t = k = 4, n = 10$ .

Optimality	Rel-Eff (to Gurobi)	Rel-Eff (rounding)	Time (sec)	Gurobi Time (sec)
A	0.9807	0.9756	1.21	2.86
D	0.9265	0.9116	5.34	2.86

**Example 5.**  $(t, p) = (4, 4), n = 12$ . Similarly, we assumed all direct treatment effects to be 1.732 and  $\lambda = 0.1$ , and the covariance structure is taken to be the identity matrix. As shown in Table 5, the efficiency of the OWEA designs relative to the Gurobi designs is 1.009 and 1.027 under A- and D-optimal criteria. There are minor differences in computing time, but all the computing times are below 0.3 seconds. In addition, design 1 (shown as  $d_1$  in Table 5) in [29] is a set of 3 mutually orthogonal Latin square, which is balanced and neighbor-balanced. Comparably, the OWEA design is almost identical in efficiency under A- and D-optimal criteria.

$$A - opt = \begin{matrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 4 \\ 2 & 3 & 3 & 1 & 3 & 4 & 2 & 4 & 4 & 1 & 1 & 2 \\ 4 & 2 & 4 & 3 & 4 & 3 & 1 & 1 & 1 & 2 & 2 & 3 \\ 3 & 4 & 2 & 4 & 1 & 1 & 4 & 2 & 2 & 3 & 3 & 1 \end{matrix} \quad (4.11)$$

$$D - opt = \begin{matrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 4 \\ 2 & 2 & 3 & 3 & 3 & 4 & 1 & 4 & 4 & 1 & 1 & 2 \\ 4 & 4 & 2 & 1 & 1 & 3 & 4 & 1 & 2 & 2 & 3 & 3 \\ 3 & 3 & 4 & 4 & 4 & 1 & 2 & 2 & 1 & 3 & 2 & 1 \end{matrix} \quad (4.12)$$

$$\text{Gurobi Design} = \begin{matrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 & 4 & 4 \\ 2 & 3 & 4 & 1 & 3 & 4 & 1 & 2 & 4 & 1 & 2 & 2 \\ 4 & 2 & 3 & 3 & 4 & 3 & 4 & 1 & 1 & 2 & 1 & 3 \\ 3 & 4 & 2 & 4 & 1 & 1 & 2 & 4 & 2 & 3 & 3 & 1 \end{matrix} \quad (4.13)$$

used as benchmarks. As it is indicated in Table 6, the relative efficiency of A- and D- optimal designs are 0.9807 and 0.9265 compared to the Gurobi designs. Note that the relatively lower D- efficiency is the limitation of the OWEA framework, especially when  $n$  is small. There is no obvious winner in computing times, as OWEA framework is faster searching for A-optimal designs but slower in D-optimal designs compared to those using Gurobi.

$$A - opt = \begin{matrix} 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 & 4 \\ 2 & 3 & 4 & 1 & 2 & 1 & 3 & 1 & 2 & 3 \\ 4 & 2 & 3 & 3 & 4 & 4 & 2 & 1 & 2 \\ 4 & 4 & 3 & 4 & 1 & 2 & 1 & 3 & 1 & 2 \end{matrix} \quad (4.14)$$

$$D - opt = \begin{matrix} 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 & 4 \\ 2 & 3 & 4 & 2 & 4 & 2 & 3 & 2 & 3 & 4 \\ 3 & 4 & 2 & 4 & 1 & 1 & 1 & 3 & 2 & 1 \\ 3 & 4 & 2 & 3 & 3 & 1 & 4 & 1 & 1 & 2 \end{matrix} \quad (4.15)$$

$$\text{Gurobi Design} = \begin{matrix} 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 & 4 & 4 \\ 1 & 2 & 1 & 2 & 3 & 1 & 3 & 1 & 3 & 4 \\ 4 & 4 & 3 & 4 & 1 & 4 & 2 & 3 & 2 & 2 \\ 4 & 3 & 4 & 1 & 1 & 2 & 1 & 3 & 2 & 3 \end{matrix} \quad (4.16)$$

**Example 7.**  $(t, k) = (4, 5), n = 24$ . Based on Theorem 6 (ii) of [13], the universally optimal design is a symmetric design comprising equal weights to sequences and their dual sequences, where the dual sequence has the same elements as its original sequence but in reversed order. The design referred as ‘‘Universal’’ in Table 7 is a design consists of symmetric blocks  $\langle 11234 \rangle, \langle 12344 \rangle$ . From the summary in Table 7, the OWEA designs are as efficient as universally

### 4.3 Interference Model

**Example 6.**  $(t, k) = (4, 4), n = 10$ . As it is shown in [11], the optimal weights of universally optimal design are not integers, and hence the Gurobi designs proposed therein are



Table 7.  $t = 4, k = 5, n = 24$ .

Optimality	Rel-Eff (to Universal)	Rel-Eff (to Gurobi)	Time (sec)	Gurobi Time (sec)
A	0.9970	0.9970	14.17	0.20
D	0.9981	0.9981	7.89	0.20

optimal designs from [13] and the Gurobi design, with the relative efficiency being 0.9970 and 0.9981 under A- and D- optimality. The computing times, in this case, are 14.17 and 7.89 seconds. Interestingly, integer programming for the Gurobi design only takes 0.20 seconds. This is because of the existence of universally optimal designs and solving for a linear equation systems is straightforward and fast utilizing integer programming.

$$A - opt = \begin{matrix} 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 & 3 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 2 & 2 & 2 & 2 & 3 & 3 & 4 & 1 & 3 & 4 & 4 & 4 & 4 \\ 2 & 3 & 3 & 3 & 4 & 4 & 1 & 3 & 3 & 1 & 1 & 3 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 3 & 2 & 2 & 2 & 2 & 3 \\ 3 & 4 & 4 & 4 & 2 & 3 & 4 & 4 & 4 & 3 & 3 & 3 & 4 & 4 & 4 & 1 & 2 & 2 & 2 & 2 & 1 & 3 & 3 & 1 & 1 & 1 \\ 4 & 2 & 2 & 2 & 3 & 3 & 3 & 1 & 1 & 3 & 3 & 1 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 2 & 2 & 1 & 1 & 1 & 1 & 2 \end{matrix} \quad (4.17)$$

$$D - opt = \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 3 & 3 & 3 & 2 & 2 & 2 & 2 & 4 & 4 & 4 & 2 & 2 & 2 & 3 & 3 & 3 & 1 & 1 & 1 & 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 & 4 & 4 & 3 & 3 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 2 & 2 & 2 & 4 & 4 & 4 & 1 & 1 & 1 & 4 & 4 & 4 & 2 & 2 & 2 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 4 & 4 & 4 & 4 & 4 & 4 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{matrix} \quad (4.18)$$

$$\text{Gurobi Design} = \begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 2 & 2 & 4 & 2 & 2 & 2 & 3 & 3 & 4 & 1 & 2 & 3 & 3 & 3 & 4 & 1 & 1 & 3 & 4 & 4 & 4 & 4 & 4 \\ 2 & 2 & 3 & 3 & 4 & 3 & 3 & 4 & 4 & 1 & 4 & 1 & 4 & 1 & 1 & 1 & 4 & 2 & 2 & 3 & 2 & 1 & 2 & 3 & 1 & 2 & 3 \\ 3 & 4 & 4 & 4 & 3 & 2 & 1 & 1 & 3 & 4 & 1 & 3 & 2 & 4 & 2 & 4 & 2 & 1 & 3 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ 4 & 3 & 2 & 4 & 3 & 2 & 4 & 3 & 1 & 4 & 1 & 3 & 2 & 4 & 4 & 2 & 1 & 1 & 3 & 2 & 1 & 3 & 3 & 1 & 1 & 1 & 1 \end{matrix} \quad (4.19)$$

## 5. DISCUSSION

In this paper, we provide an algorithm framework for optimal or efficient crossover designs, and later extend it to the interference model. Information matrices are derived in order to apply the optimal weight exchange algorithm. The OWEA exact designs in the numerical examples are shown to be efficient or identical to the optimal exact designs. The most noteworthy advantages of the OWEA algorithm are the generality and convenience. It works on a myriad of models, as long as the information matrix is linear with respect to the design point. Although the requirement on the information matrix is regarded as a limitation of the algorithm, in practice, it is typically met when using linear models or some non-linear model. From one model to another, the OWEA algorithm is applicable with only minor modifications to the information matrix. Within the same model, one only needs to adjust the values of case configurations, such as  $t$ ,  $p$ , and  $n$  in those examples. Note that these configurations can be arbitrary, including the inclusion of a dropout mechanism. To facilitate the usage of our algorithm, we have prepared an R package and provided an online R Shiny app as a more user-friendly interface (<https://cran.r-project.org/package=OWEA>).

Although the theoretical result indicates optimal designs within a subclass of pseudo-symmetric designs are automatically global optimal, the OWEA algorithm does not guarantee symmetry in its output. However, based on our experience, OWEA exact designs are close to symmetric optimal designs and always have relatively high efficiency or satisfactory lower bound. In addition, we also observed the limitation of the framework when  $n$  is a small integer, where the efficiency is not very high. In our opinion, one of the reasons is that approximate optimal design usually consists of many distinct sequences with small weights, and rounding them into a small  $n$  exact design would result in losing many sequences that significantly contributes to the information matrix. Furthermore, the OWEA algorithm offers flexibility in the choice of parameter vectors. Note that all the designs in this paper focus on direct treatment effects. In other cases, for example, [25] derived corresponding results for estimating  $\lambda$  in the proportional model, whereas using OWEA algorithm, one only needs to change the function  $g$ .

Speed is also an important aspect of numerical algorithms. In the numerical examples, the OWEA algorithm only takes a reasonable amount of time to complete the tasks. However, there is still space for improvement. The current search spaces that we implemented in the numerical examples comprise almost all possible sequences or blocks. According to theory of Kushner’s approximate design theory, for example, [11, 13, 25], one can narrow the search space down to a much smaller one depending on combinations of  $(t, p, n)$  or  $(t, k, n)$ . However, there is a trade-off between convenience and computing time. Another possible improvement lies in the Newton’s method. In some of the cases, we do encounter overshooting problems, but the resultant design after reaching maximum iterations is still highly efficient. If this issue could be resolved, the computing time would be faster than the reported time.

Finally, although we compared examples and algorithms in [9, 13, 25], one has to point out those papers are aiming at developing powerful theoretical tools while this paper is prone to be more practical oriented. Often, practitioners even statisticians cannot easily understand and use good theoretical results in design studies. This inability to use results greatly hinders the application of excellent theoretical work. The developed R package R Shiny app not only offers a user-friendly interface but also generates optimal and efficient designs that are not available in existing publications. Deriving such designs theoretically is often challenging or impossible, but our user-friendly package simplifies the process. This development is essential for making theoretical results more accessible to practitioners, and it will

undoubtedly contribute to the continued advancement of this important field.

## ACKNOWLEDGMENTS

We would like to extend our sincere appreciation to the editors for their dedicated efforts in reviewing our research paper. Their insightful comments and suggestions have undoubtedly contributed to the overall improvement of the manuscript.

## FUNDING

Min Yang was supported by NSF Grant DMS-22-10546.

*Accepted 9 June 2023*

## REFERENCES

- [1] YANG, M., BIEDERMANN, S. and TANG, E. (2013). On optimal designs for nonlinear models: a general and efficient algorithm. *Journal of the American Statistical Association* **108**(504) 1411–1420. <https://doi.org/10.1080/01621459.2013.806268>. MR3174717
- [2] YANG, M. and STUFKEN, J. (2008). Optimal and efficient crossover designs for comparing test treatments to a control treatment under various models. *Journal of Statistical Planning and Inference* **138**(1) 278–285. arXiv:0505641. <https://doi.org/10.1016/j.jspi.2007.05.005>. MR2369633
- [3] LOW, B. Y. J. L., LEWIS, S. M. and PRESCOTT, P. (1999). Assessing robustness of crossover designs to subjects dropping out. *Statistics and Computing* **9**(3) 219–227. <https://doi.org/10.1023/A:1008974031782>
- [4] MATTHEWS, J. N. S. (1988). Recent developments in crossover designs. *International Statistical Review / Revue Internationale de Statistique* **56**(2) 117. <https://doi.org/10.2307/1403636>. MR0963525
- [5] GODOLPHIN, J. D. (2004). Simple pilot procedures for the avoidance of disconnected experimental designs. *Journal of the Royal Statistical Society. Series C: Applied Statistics* **53**(1) 133–147. <https://doi.org/10.1046/j.0035-9254.2003.05054.x>. MR2043764
- [6] MAJUMDAR, D., DEAN, A. M. and LEWIS, S. M. (2008). Uniformly balanced repeated measurements designs in the presence of subject dropout. *Statistica Sinica* **18** 235–253. arXiv:0710.0685. MR2384987
- [7] BOSE, M. and BAGCHI, S. (2008). Optimal crossover designs under premature stopping. *Utilitas Mathematica* **75** 273–285. MR2392763
- [8] ZHAO, S. and MAJUMDAR, D. (2012). On uniformly balanced crossover designs efficient under subject dropout. *Journal of Statistical Theory and Practice* **6**(1) 178–189. <https://doi.org/10.1080/15598608.2012.647579>. MR3196538
- [9] ZHENG, W. (2013). Universally optimal crossover designs under subject dropout. *The Annals of Statistics* **41**(1) 63–90. arXiv:1303.2800v1. <https://doi.org/10.1214/12-AOS1074>
- [10] KUSHNER, H. B. (1997). Optimal repeated measurements designs: the linear optimality equations. *The Annals of Statistics* **25**(6) 2328–2344. <https://doi.org/10.1214/aos/1030741075>. MR1604457
- [11] KUNERT, J. and MARTIN, R. J. (2000). On the determination of optimal designs for an interference model. *The Annals of Statistics* **28**(6) 1728–1742. <https://doi.org/10.1214/aos/1015957478>. MR1835039
- [12] KUNERT, J. and MERSMANN, S. (2011). Optimal designs for an interference model. *Journal of Statistical Planning and Inference* **141**(4) 1623–1632. <https://doi.org/10.1016/j.jspi.2010.11.020>. MR2747931
- [13] ZHENG, W. (2015). Universally optimal designs for two interference models. *Ann. Statist.* **43**(2) 501–518. <https://doi.org/10.1214/14-AOS1287>
- [14] FEDOROV, V. (1972) *Theory of Optimal Experiments Designs*. New York: Academic Press. MR0403103
- [15] WYNN, H. P. (1970). The sequential generation of D-optimum experimental designs. *The Annals of Mathematical Statistics* **41**(5) 1655–1664. <https://doi.org/10.1214/aoms/1177696809>
- [16] SILVEY, S. D., TITTERINGTON, D. H. and TORSNEY, B. (1978). An algorithm for optimal designs on a design space. *Communications in Statistics – Theory and Methods* **7**(14) 1379–1389. <https://doi.org/10.1080/03610927808827719>
- [17] TORSNEY, B. Algorithms for a constrained optimization problem with applications in statistics and optimum design (1981). PhD thesis, University of Glasgow.
- [18] HETTICH, R. (1983). A review of numerical methods for semi-infinite optimization. In *Semi-Infinite Programming and Applications* (A. V. Fiacco and K. O. Kortanek, eds.) 158–178. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [19] BÖHNING, D. (1986). A vertex-exchange-method in D-optimal design theory. *Metrika* **33**(1) 337–347. <https://doi.org/10.1007/BF01894766>. MR0868043
- [20] HARMAN, R. and PRONZATO, L. (2007). Improvements on removing nonoptimal support points in D-optimum design algorithms. *Statistics and Probability Letters* **77**(1) 90–94. arXiv:0706.4394. <https://doi.org/10.1016/j.spl.2006.05.014>. MR2339022
- [21] TORSNEY, B. and MARTÍN-MARTÍN, R. (2009). Multiplicative algorithms for computing optimum designs. *Journal of Statistical Planning and Inference* **139**(12) 3947–3961. <https://doi.org/10.1016/j.jspi.2009.05.007>. MR2558341
- [22] MARTÍN-MARTÍN, R., RODRÍGUEZ-ARAGÓN, L. J. and TORSNEY, B. (2012). Multiplicative algorithm for computing D-optimum designs for pVT measurements. *Chemometrics and Intelligent Laboratory Systems* **111**(1) 20–27. <https://doi.org/10.1016/j.chemolab.2011.10.014>. MR2558341
- [23] YU, Y. (2010). Monotonic convergence of a general algorithm for computing optimal designs. *The Annals of Statistics* **38**(3) 1593–1606. arXiv:0905.2646. <https://doi.org/10.1214/09-AOS761>
- [24] KIEFER, J. (1974). General equivalence theory for optimum designs (approximate theory). *The Annals of Statistics* **2**(5) 849–879. <https://doi.org/10.1214/aos/1176342810>. MR0356386
- [25] ZHENG, W. (2013). Optimal crossover designs for the proportional model. *The Annals of Statistics* **41**(4) 2218–2235. <https://doi.org/10.1214/13-AOS1148>. MR3127864
- [26] GUROBI OPTIMIZATION, LLC (2022). *Gurobi Optimizer Reference Manual*. <https://www.gurobi.com>.
- [27] R CORE TEAM (2022). R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- [28] WANG, X. and CHINCHILLI, V. M. (2021). Analysis of crossover designs with nonignorable dropout. *Statistics in Medicine* **40**(1) 64–84. <https://doi.org/10.1002/sim.8762>. MR4194572
- [29] KEMPTON, R. A., FERRIS, S. J. and DAVID, O. (2001). Optimal change-over designs when carry-over effects are proportional to direct effects of treatments. *Biometrika* **88**(2) 391–399. <https://doi.org/10.1093/biomet/88.2.391>. MR1844839

Shuai Hao. AbbVie Inc, USA.

E-mail address: [shuai.hao@abbvie.com](mailto:shuai.hao@abbvie.com)

Min Yang. Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, USA.

E-mail address: [myang2@uic.edu](mailto:myang2@uic.edu)

Wei Zheng. Department of Business Analytics and Statistics, University of Tennessee-Knoxville, USA.

E-mail address: [wzheng9@utk.edu](mailto:wzheng9@utk.edu)