1-1-2013

On Optimal Designs for Nonlinear Models: A General and Efficient Algorithm

Min Yang

Stefanie Biedermann

Yihui Tang

Original Citation

Yang, Min, Stefanie Biedermann and Yihui Elina Tang (2013), "On Optimal Designs for Nonlinear Models: A General and Efficient Algorithm", Journal of the American Statistical Association, 108 (504), 1411 – 1420.

Journal of the American Statistical Association

ISSN: 0162-1459 (Print) 1537-274X (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

On Optimal Designs for Nonlinear Models: A General and Efficient Algorithm

Min Yang , Stefanie Biedermann & Elina Tang

To cite this article: Min Yang , Stefanie Biedermann & Elina Tang (2013) On Optimal Designs for Nonlinear Models: A General and Efficient Algorithm, Journal of the American Statistical Association, 108:504, 1411-1420, DOI: 10.1080/01621459.2013.806268

To link to this article: https://doi.org/10.1080/01621459.2013.806268

Accepted author version posted online: 09 Jun 2013.
Published online: 19 Dec 2013.

On Optimal Designs for Nonlinear Models: A General and Efficient Algorithm

Min YANG, Stefanie BIEDERMANN, and Elina TANG

Finding optimal designs for nonlinear models is challenging in general. Although some recent results allow us to focus on a simple subclass of designs for most problems, deriving a specific optimal design still mainly depends on numerical approaches. There is need for a general and efficient algorithm that is more broadly applicable than the current state-of-the-art methods. We present a new algorithm that can be used to find optimal designs with respect to a broad class of optimality criteria, when the model parameters or functions thereof are of interest, and for both locally optimal and multistage design strategies. We prove convergence to the optimal design, and show in various examples that the new algorithm outperforms the current state-of-the-art algorithms.

KEY WORDS: Convergence; Locally optimal design; Multistage design; Φ_p-optimality.

1. INTRODUCTION

Experimental design is an integral part of scientific research. An optimal or efficient design, by using resources economically, can facilitate the data-collection process and subsequent data analysis, thus leading to reliable and reproducible conclusions in a cost-effective manner. Much design work thus far focuses on linear models. However, many natural phenomena follow nonlinear models. Efficient designs for nonlinear models are needed in a multitude of application areas.

A major complication in studying optimal designs for nonlinear models is that information matrices and thus optimal designs depend on the unknown model parameters, held in the vector $\boldsymbol{\theta}=(\theta_1, \ldots, \theta_k)'$. A typical approach is to use a locally optimal design, which is based on a "best guess" of the unknown parameters $\boldsymbol{\theta}$ (Chernoff 1953). One would first make a "guess" about the value of $\boldsymbol{\theta}$, and then start the search for an optimal design accordingly. Such an approach will inevitably run into a hit-and-miss problem. Many times, one may get lucky and obtain a good guess of $\boldsymbol{\theta}$. Other times, the initial guess of $\boldsymbol{\theta}$ is far from the true value and the resulting locally optimal design is far from the true optimal design. A practical way to get a reliable "best guess" of $\boldsymbol{\theta}$ is to employ response adaptive or multistage designs, an approach that has gained a lot of popularity in practice in the past decade. An initial experiment, typically using a robust design, is conducted to get a better idea about the unknown parameters. The resulting initial estimate for $\boldsymbol{\theta}$ is then used as the "best guess" of $\boldsymbol{\theta}$, based on which the next stage design is selected. The research question here is to find a design such that the combination of the initial design and the second-stage design is optimal/efficient with respect to the selected optimality criterion. The observations from both the initial and the second-stage designs are subsequently used to obtain new estimates for the parameters. If a third-stage design is needed, these will serve as the "best guess," and so on. Such multistage design strategies have recently been used by pharmaceutical companies in clinical trial experimental designs. For example, a three-stage design was used in a dose-response study for a new drug treating acute pain (Dragalin, Hsuan, and Padmanabhan 2007).

Recently, Yang and Stufken (2009), Yang (2010), and Dette and Melas (2011) obtained a series of unifying results for a large class of nonlinear models, multiple optimality criteria, and multiple objectives. They showed that we can focus on a subclass of designs with a simple form, which is dominating in the Loewner sense, implying that for each design and optimality criterion there is a design in the subclass that is at least as good. While these results are big steps toward simplifying design search for nonlinear models, the numerical computation may still be problematic. For example, suppose the dominating class consists of designs with at most five support points. Then we still have nine variables (five support points and four weights) to be determined. Obviously, a full grid search is not feasible in this situation, and efficient algorithms are needed.

There are several algorithms available in the literature, most of which are modifications of either the Fedorov-Wynn algorithm (FWA; Wynn 1970; Fedorov 1972) or the multiplicative algorithm (MA; Silvey, Titterington, and Torsney 1978). The FWA is concerned with updating the support of the design, whereas the MA operates with updating the design weights only. Optimization of weights works as a first-order optimization procedure. Therefore, these two algorithms are asymptotically slow and any algorithm optimizing both points and weights will be faster (see, e.g., Torsney 1981; Hettich 1983, for several-phase approaches to determine optimal designs). As a result, many different modifications of these algorithms have been proposed (e.g., Böhning 1986; Harman and Pronzato 2007; Torsney and Martín-Martín 2009; Martín-Martín, Rodríguez-Aragón, and Torsney 2012). However, these modifications mainly focus on one specific design problem, D-optimality, when all parameters are of interest. Chernoff (1999) expressed concerns about the

Min Yang is Professor, Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, IL 60607 (E-mail: *myang2@uic.edu*). Stefanie Biedermann is Lecturer, School of Mathematics, University of Southampton, Highfield, Southampton, SO17 1BJ (E-mail: *S.Biedermann@soton.ac.uk*). Elina Tang is Assistant Professor, Department of Managerial Studies, University of Illinois at Chicago, Chicago, IL 60607 (E-mail: *etang@uic.edu*). The authors are thankful for detailed comments and suggestions by the editor, the associate editor, and three referees on an earlier version of the article. Yang's research was supported by NSF grants DMS-0707013, DMS-1322797, and FDA MCM Challenge Grant. Tang's research was partially supported by the Donald W. Reynolds Journalism Institute. The contents are solely the responsibility of the authors and do not necessarily represent the official views of FDA.

Journal of the American Statistical Association
December 2013, Vol. 108, No. 504, Theory and Methods
DOI: 10.1080/01621459.2013.806268

concentration on D-optimality in the literature, advocating optimality results under different criteria. In particular, the selection of an appropriate optimality criterion depends on the main objective of the experiment. For example, if there are nuisance parameters in the model, we may want to choose a design that is optimal for estimating the parameters of interest only. In addition, and perhaps even more important, these algorithms cannot be used for deriving optimal multistage designs. An exception is Covey-Crump and Silvey (1970) who found two-stage D- and E-optimal designs for multivariable polynomial models.

Recently, Yu (2011) proposed a new algorithm named "Cocktail algorithm" (CA), which combines in each step an iteration of the FWA, a nearest neighbor exchange following the VEM, and an iteration of the MA, in a way that increases speed considerably compared with each individual method. However, it is restricted to D-optimal designs for the full parameter vector $\boldsymbol{\theta}$, and may not be directly applied to find multistage designs. These are major obstacles to wider use of the optimal design approach by practitioners.

The purpose of this article is to develop a general yet efficient algorithm that can address this gap in the literature. We propose a new algorithm, optimal weights exchange algorithm, that is applicable to a large class of optimality criteria, and covers the situations where a subset or a (differentiable) function of the parameters is of interest. Furthermore, our algorithm finds not only locally optimal designs, but also multistage designs. Our method can be viewed as an extension of the FWA by adding an optimization step for the weights. Specifically, we start with a nonsingular design, optimize the weights for the current support, and remove points with zero weights. Next, we add one point to the support (as in the FWA), and optimize the weights again, and so on and so forth. We propose to optimize the weights using the Newton method, a second-order optimization method, which increases the speed.

We investigate the theoretical properties of the new algorithm, and prove convergence in many practical situations. Silvey (1980) noted: "What is important about an algorithm is not whether it converges, but whether it is effective in the sense that it guarantees arbitrary close approach to the optimum; and how fast this approach is." We show that the new algorithm is highly efficient for all different optimality problems. In fact, for those problems to which the CA can be applied, the new algorithm outperforms the CA by a large scale.

In addition, we investigate how to select a grid to substitute a continuous design space. It is common practice to consider a design set with grid points spread equidistantly in each variable. The finer the grid, the better the design obtained, but the higher the computational burden, especially in higher dimensions. We derive a lower bound for the efficiency of a design that is optimal on a grid, relative to the corresponding optimal design on the continuous design space. This helps us to determine how fine the grid should be to avoid unnecessary computational effort.

This article is organized as follows. In Section 2, we introduce the necessary notations. The main results including convergence properties, implementation of the algorithm, as well as efficiency considerations in continuous space are presented in Section 3. Applications to many commonly studied nonlinear models, and comparisons with the current state-of-the-art algorithms are shown in Section 4. Section 5 provides a brief discussion, followed by an Appendix containing the proofs.

2. SETUP AND NOTATIONS

2.1 The Design Problem

Suppose we have a nonlinear regression model for which at each point $\mathbf{x}$ the experimenter observes a response Y. Here $\mathbf{x}$ could be a vector, and we assume that the responses are independent and follow some distribution from the exponential family with mean $\eta(\mathbf{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the $(k \times 1)$ vector of unknown parameters. Typically, approximate designs are studied, that is, designs of the form $\xi = \{(\mathbf{x}_i, \omega_i), i = 1, \ldots, m\}$ with support points $\mathbf{x}_i \in \mathcal{X}$ and weights $\omega_i > 0$, and $\sum_{i=1}^{m} \omega_i = 1$. Let $\mathcal{X}$ represent the set of all possible design points. For a numerical study, while the original design space $\mathcal{C}$ is usually continuous, we consider $\mathcal{X}$ to be a set of grid points spread equidistantly in each variable.

In the multistage design context, let ξ_0 denote the design we have already carried out, and n_0 be the number of observations. Note that ξ_0 is an exact design, that is, a special case of an approximate design where the weights multiplied with n_0 yield integers, the exact number of observations at each support point. Suppose we can take n_1 observations at the next stage, so we need to determine the design ξ such that the combined design $\xi_0 + \xi$ optimizes the selected optimality criterion. The operator "+" means that the support of the combined design consists of the support points of ξ_0 and ξ, where the corresponding weights are weighted averages of their original weights. For example, if $\mathbf{x}$ is a support point of both designs, with weights w_0 and w, respectively, its weight in $\xi_0 + \xi$ will be $w_0 n_0/(n_0 + n_1) + w n_1/(n_0 + n_1)$. For ease of computation, the design ξ is an approximate design, that is, its weights will have to be rounded appropriately before the design can be applied.

Denote the information matrix at a single point $\mathbf{x}$ as $\mathbf{I}_{\mathbf{x}}$. The information matrix of a design ξ can then be written as $\mathbf{I}_{\xi} = \sum_{i=1}^{m} \omega_i \mathbf{I}_{\mathbf{x}_i}$, and the information matrix of the combined design $\xi_0 + \xi$ is $\mathbf{I}_{\xi_0+\xi} = n_0/(n_0 + n_1)\mathbf{I}_{\xi_0} + n_1/(n_0 + n_1)\mathbf{I}_{\xi}$.

Throughout this article, we assume that $\mathbf{I}_{\xi_0}$ is nonsingular and $n_0 > 0$ unless specified otherwise. Such assumptions are common in design for nonlinear models, especially for an algorithmic approach. In fact, we expect the information matrix of the initial design to be nonsingular, since the purpose of an initial design is to obtain estimates for all parameters.

Let $g(\boldsymbol{\theta}) = (g_1(\boldsymbol{\theta}), \ldots, g_v(\boldsymbol{\theta}))^T$, $1 \le v \le k$, be the (possibly vector-valued) differentiable function of the parameters, which is of interest. We can estimate $g(\boldsymbol{\theta})$ using the maximum likelihood estimator $g(\hat{\boldsymbol{\theta}})$, where $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimator of $\boldsymbol{\theta}$. The asymptotic variance-covariance matrix of $g(\hat{\boldsymbol{\theta}})$ under design $\xi_0 + \xi$ can be written as

$$\boldsymbol{\Sigma}_{\xi_0+\xi}(g) = \frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \mathbf{I}_{\xi_0+\xi}^{-1} \left(\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \right)^T.$$

The aim is to identify a design ξ, such that the variance-covariance matrix $\boldsymbol{\Sigma}_{\xi_0+\xi}(g)$ is minimized under the selected optimality criterion.

There are a variety of optimality criteria. The commonly used ones are A-, D-, and E-optimality, which is to minimize $Tr(\boldsymbol{\Sigma}_{\xi_0+\xi}(g))$, $|\boldsymbol{\Sigma}_{\xi_0+\xi}(g)|$, and $\lambda_{\max}$, respectively, where $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{\Sigma}_{\xi_0+\xi}(g)$. These optimality criteria are appealing because of their statistical meanings. For example, an A-optimal design minimizes the sum of the variances of the estimators, a D-optimal design minimizes the volume of

the confidence ellipsoid of the estimators, and an E-optimal design protects against the worst case for inference. Kiefer (1974), in an effort to unify these criteria, defined the class of functions

$$\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi}(g)) = \left[\frac{1}{v}\mathrm{Tr}\,(\mathbf{\Sigma}_{\xi_0+\xi}(g))^p\right]^{1/p},\ 0 \le p < \infty.$$

The case $p = 0$ is understood as the limit $\lim_{p\to 0}\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi}(g)) = |\mathbf{\Sigma}_{\xi_0+\xi}(g)|^{1/v}$ (D-optimality); for $p = 1$, we have A-optimality; $\lim_{p\to\infty}\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi}(g)) = \lambda_{\max}$, and we obtain E-optimality for $p \to \infty$. Throughout this article, we shall consider Φ_p-optimality. For technical reasons, we restrict to p being a nonnegative integer. This restriction has little impact on any practical optimality problem since it is rare to consider Φ_p-optimality for noninteger p. Note that minimizing $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))$ is equivalent to minimizing

$$\widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g)) = \begin{cases} \log|\mathbf{\Sigma}_{\xi_0+\xi}(g)|, & \text{if } p = 0; \\ \mathrm{Tr}(\mathbf{\Sigma}_{\xi_0+\xi}(g))^p, & \text{if } p > 0. \end{cases}$$

A powerful tool for checking the optimality of a given design is an equivalence theorem. We thus present the well-known equivalence theorem for Φ_p-optimal multistage designs. All designs in this article have been carefully checked using Theorem 1.

Theorem 1. In a multistage design, suppose that $n_0 > 0$ and $\mathbf{I}_{\xi_0}$ is nonsingular. Let the directional derivative of Φ_p, $d_p(\mathbf{x}, \xi)$, be defined as

$$d_p(\mathbf{x},\xi) = \begin{cases} \mathrm{Tr}((\mathbf{\Sigma}_{\xi_0+\xi}(g))^{-1}A(g,\xi)), & \text{if } p = 0; \\ \left(\frac{1}{v}\right)^{\frac{1}{p}}(\mathrm{Tr}(\mathbf{\Sigma}_{\xi_0+\xi}(g))^p)^{\frac{1}{p}-1} & \\ \quad \mathrm{Tr}((\mathbf{\Sigma}_{\xi_0+\xi}(g))^{p-1}A(g,\xi)), & \text{if } p > 0; \end{cases} \quad (1)$$

where

$$A(g,\xi) = \frac{n_1}{n_0+n_1}\left(\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right)(\mathbf{I}_{\xi_0+\xi})^{-1}(\mathbf{I}_{\mathbf{x}} - \mathbf{I}_{\xi})(\mathbf{I}_{\xi_0+\xi})^{-1} \times \left(\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right)^T.$$

Then a design ξ^* is Φ_p-optimal for $g(\boldsymbol{\theta})$ if and only if, for all $\mathbf{x} \in \mathcal{X}$,

$$d_p(\mathbf{x}, \xi^*) \le 0,$$

with equality if $\mathbf{x}$ is any support point of a Φ_p-optimal design.

2.2 Notations and Idea

The proposed new algorithm requires a starting set of initial points $S^{(0)}$. Let $S^{(t)}$ denote the set of support points at the tth iteration, and $\xi_{S^{(t)}}$ denote the design with support points $S^{(t)}$ with optimal weights. In each step of the algorithm, one further grid point is added to the current support $S^{(t)}$, and the design $\xi_{S^{(t+1)}}$ is then found by directly optimizing the weights for the new support $S^{(t+1)}$. This optimization may yield an optimum on the boundary, that is, one or more weights may be zero, in which case the corresponding support points are deleted from $S^{(t+1)}$. The updating rule for the support is given by

$$S^{(t+1)} = S^{(t)} \bigcup \{\mathbf{x}_t^*\}, \quad \text{where } \mathbf{x}_t^* = \arg\max_{\mathbf{x}\in\mathcal{X}} d_p(\mathbf{x}, \xi_{S^{(t)}}). \quad (2)$$

Note that the updating rule for the support in Equation (2) is the same as in the FWA.

3. CONVERGENCE AND OPTIMAL WEIGHTS

We first establish convergence of the new algorithm when a multistage design is sought.

Theorem 2. Suppose that $n_0 > 0$, and that $\mathbf{I}_{\xi_0}$ is nonsingular. Let $\frac{\partial g}{\partial \boldsymbol{\theta}^T}$ be a matrix of full row rank. For any set of initial points $S^{(0)}$, the sequence of designs $\{\xi_{S^{(t)}}; t \ge 0\}$ converges to an optimal design that minimizes $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))$ as $t \to \infty$.

The convergence result of Theorem 2 also holds when $n_0 = 0$, that is, for one-stage or locally optimal designs, if an additional condition is satisfied.

Theorem 3. Suppose that $n_0 = 0$ and that the initial set $S^{(0)}$ satisfies $\mathbf{I}_{\xi_{S^{(0)}}} > 0$. Furthermore, let $\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}$ be a square matrix of full rank. Then, as $t \to \infty$, the sequence of designs $\{\xi_{S^{(t)}}; t \ge 0\}$ converges to an optimal design that minimizes $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))$.

If $g(\boldsymbol{\theta}) = \boldsymbol{\theta}$, then $\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}$ is the $(k \times k)$-identity matrix. Theorem 3 can thus be applied to any locally Φ_p-optimal design for the full parameter vector $\boldsymbol{\theta}$. This includes the well-studied D-optimality problem for all parameters.

A critical step in the proposed algorithm is to find the optimal weights for given support points. Pukelsheim and Torsney (1991) gave an explicit formula for finding the optimal weights. Although this formula was presented in the context of linear models, it can be extended to nonlinear models, in which case the weights depend on the model parameters. Their approach, however, has two limitations. First, it requires the regression vectors to be independent. As a result, the number of support points cannot be greater than the number of parameters. For example, when one searches for the optimal weights for $S^{(t+1)}$, which consists of at least $k + 1$ support points (support of $\xi_{S^{(t)}}$ and $\mathbf{x}_t^*$), the number of support points is greater than the number of parameters and the above formula will not work. Second, this formula is specifically developed for one-stage designs, and cannot be directly extended to multistage designs, which is the focus of this article.

To derive the optimal weights for both one-stage and multistage designs efficiently, we propose a direct approach. Theorem 4 provides a property of the optimal weights for given support points, which facilitates their numerical computation. Let $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_{m-1})^T$. Denote $\Omega = \{\boldsymbol{\omega} : \omega_i \ge 0,\ i = 1, \ldots, m-1, \sum_{i=1}^{m-1}\omega_i \le 1\}$.

Theorem 4. In a multistage design, suppose that $\mathbf{I}_{\xi_0}$ is nonsingular and $n_0 > 0$. For given $\boldsymbol{\theta}$ and support points $(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ of ξ, $\widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))$ is minimized at any critical point in Ω (i.e., the points where $\frac{\partial\widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))}{\partial\boldsymbol{\omega}} = 0$, the zero-vector), or at the boundary of Ω, that is, $\omega_i = 0$ for some $1 \le i \le m$. In addition, the Hessian matrix of $\widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))$, $\frac{\partial^2\widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))}{\partial\boldsymbol{\omega}\partial\boldsymbol{\omega}^T}$, is a nonnegative definite matrix.

Note that when $p > 0$, we can obtain a more general result if we replace $\widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))$ by $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))$ in Theorem 4. The proof can be derived directly utilizing the convexity of Φ_p. However, the corresponding Hessian matrix of $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))$

is more cumbersome to handle in computations. The proof of Theorem 4 also provides the gradient and the Hessian matrix of $\widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))$, which are needed for numerical search.

From Theorem 4, we need to solve $m-1$ nonlinear equations to find optimal weights for given support points. In general there is no closed form solution, so we employ a numerical approach. Newton's method features a quadratic convergence rate, which, loosely speaking, means that the number of significant digits doubles after each iteration (Isaacson and Keller 1966).

By Theorem 4, the Hessian matrix is nonnegative definite in Ω. This guarantees convergence given the starting point is sufficiently close to the critical point (Kaplan 1999). Since Ω is a compact set, we can always find the critical points (given there exists a critical point within Ω) if we use sufficiently many different initial points.

If the numerical search leads to a minimum on the boundary, we remove the design point with zero weight, and then search for the optimal weights on the reduced set of support points. This process is repeated until we find weights that satisfy the constraints, which is guaranteed since the number of support points is finite.

One may be tempted to apply Theorem 4 directly to the whole design set $\mathcal{X}$. However, this is not feasible in general, unless the size of the design set is very small, say, about the same as the number of parameters. Even for a moderate number of given support points, the computation of the Hessian matrix is time consuming, and there will be too many different boundary situations to be considered.

3.1 Implementation of the Algorithm

Theorems 2, 3, and 4 provide the theoretical foundation for the algorithm defined in Equation (2). A step-by-step procedure for its implementation in programming is described in what follows:

(i) Let $t=0$, and let $S^{(0)}$ be a set of $k+1$ design points uniformly distributed in $\mathcal{X}$ (the initial weights are uniform).
(ii) Derive optimal weights for $S^{(t)}$ using Newton's method.
(iii) Derive $\mathbf{x}_t^* = \arg\max_{\mathbf{x}\in\mathcal{X}} d_p(\mathbf{x}, \xi_{S^{(t)}})$.
(iv) Select a small value $\varepsilon_0 > 0$. If $d_p(\mathbf{x}_t^*, \xi_{S^{(t)}}) \le \epsilon_0$, $\xi_{S^{(t)}}$ is the desired design.
(v) Otherwise, set $S^{(t+1)} = S^{(t)} \bigcup \{\mathbf{x}_t^*\}$, set $t = t+1$ and repeat Steps (ii)–(iv). The optimal weights from $\xi_{S^{(t)}}$ (zero weight for $\mathbf{x}_t^*$) serve as initial weights of $S^{(t+1)}$ in Step (ii).

Here, Newton's method, for a given set of support points $S^{(t)}$ and the associated initial weights $\boldsymbol{\omega}_0^{(t)}$, updates $\boldsymbol{\omega}_j^{(t)}$, the weights after the jth iteration, as follows (starting with $\alpha = 1$).

(a) $\boldsymbol{\omega}_j^{(t)} = \boldsymbol{\omega}_{j-1}^{(t)} - \alpha\left(\frac{\partial^2 \widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))}{\partial \boldsymbol{\omega}\boldsymbol{\omega}^T}\Big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_{j-1}^{(t)}}\right)^{-1} \frac{\partial \widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))}{\partial \boldsymbol{\omega}}\Big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_{j-1}^{(t)}}$.
(b) Check if there are nonpositive components of $\boldsymbol{\omega}_j^{(t)}$. If so, go to Step (c2), otherwise proceed to Step (c1).
(c) Check whether $\left|\left|\frac{\partial \widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))}{\partial \boldsymbol{\omega}}\Big|_{\boldsymbol{\omega}=\boldsymbol{\omega}_j^{(t)}}\right|\right|$ is less than a prespecified $\tilde{\epsilon} > 0$. If so, $\boldsymbol{\omega}^{(t+1)}$ is the vector of optimal weights. Otherwise, start the next iteration.
(d) Reduce α to $\alpha/2$. Repeat Steps (a) and (b) until α reaches a prespecified value, say 0.00001. Remove the support point with smallest weight. For the new set of support points as well as their weights, go to Step (a).

Concrete expressions for $\frac{\partial \widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))}{\partial \boldsymbol{\omega}}$ and $\frac{\partial^2 \widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))}{\partial \boldsymbol{\omega}\boldsymbol{\omega}^T}$, respectively, can be found in the Appendix (formulas (A.17) and (A.18) for $p=0$, and (A.19) and (A.20) for $p>0$).

In what follows, we briefly discuss the practical properties of this algorithmic procedure.

1. Computation time. The computation time of Newton's method cannot be judged by the number of iterations only. Each iteration includes the calculation of the second derivative and evaluation of the Hessian matrix, which can be time consuming if the number of given support points in $S^{(t)}$ is large. The algorithm adds one point to the existing support in each iteration. However, for a large support at least one of the optimal weights will lie on the boundary, and the support point with zero weight will be removed, thus reducing the size of the support for the next iteration. From Caratheodory's theorem, there exists an optimal design with at most $k(k+1)/2$ support points. In a series of recent papers in this field, including those by Yang (2010), Dette and Melas (2011), and Yang and Stufken (2012), it has been demonstrated that for a large classes of nonlinear models the number of support points can be reduced to k. These results give some theoretical justification that, when the iteration progresses toward an optimal design, it is expected that the number of support points is close to k. This has been verified in our numerical studies in the next section.
2. Choice of ϵ_0. From Theorem 1, when $d_p(\mathbf{x}_t^*, \xi_{S^{(t)}}) = 0$, $\xi_{S^{(t)}}$ is Φ_p-optimal. In numerical computations, it is rare to achieve this bound. Typically we choose a small positive value, say ϵ_0, as the cut-off point, which depends on how efficient the derived design should be compared with the true optimal design from theory. From the proof of Theorem 2, $\widetilde{\Phi}_0(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) - \widetilde{\Phi}_0(\mathbf{\Sigma}_{\xi_0+\xi^*}(g)) \le d_0(\mathbf{x}_t^*, \xi_{S^{(t)}})$, and $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) - \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi^*}(g)) \le d_p(\mathbf{x}_t^*, \xi_{S^{(t)}})$ for $p>0$. These inequalities give a lower bound for the efficiency of the derived design, which is $\exp(-\frac{\epsilon_0}{v})$ for $p=0$ and $1-\frac{\epsilon_0}{\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g))}$ for $p>0$. Here, efficiency of a design ξ is defined as $\frac{\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi^*})}{\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi})}$.
3. There is no guarantee that the above Newton iteration procedure always finds the optimal weights. For all designs obtained by our procedure, we have used Theorem 1 to check whether we have indeed obtained an optimal design. In our experience, we virtually always have.
4. E-optimality is equivalent to Φ_p-optimality when $p\to\infty$, but we cannot use $p=\infty$ in practice. We propose to select a large value of p such that the corresponding Φ_p-optimal design is guaranteed to achieve at least some predescribed E-efficiency. A lower bound for the E-efficiency of such a design is found in what follows. Let ξ_p^* be a Φ_p-optimal design, $\lambda_{\max}$ be the largest eigenvalue of $\mathbf{\Sigma}_{\xi_0+\xi_p^*}(g)$, ξ_E^* be an E-optimal design, and $\lambda_{\max}^E$ be the largest eigenvalue of $\mathbf{\Sigma}_{\xi_0+\xi_E^*}(g)$. Clearly, $(\frac{1}{v})^{\frac{1}{p}}\lambda_{\max} \le \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_p^*}(g)) \le \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_E^*}(g)) \le \lambda_{\max}^E$, which gives a

lower bound for the E-efficiency of ξ_p^* as $\text{eff}_E(\xi_p^*) = \frac{\lambda_{\max}^E}{\lambda_{\max}} \geq (\frac{1}{v})^{\frac{1}{p}}$, and we choose p such that $(\frac{1}{v})^{\frac{1}{p}}$ is as close to 1 as desired. For example, if $v = 3$, we need $p = 11$ to have a lower bound of 0.905 for the E-efficiency. A large value of p may result in computational difficulties since the elements of some of the matrices are huge, leading to imprecise inverses due to rounding errors. Hence, extra care must be taken when p is relatively large. Based on our experience, the algorithm works well when $p \leq 6$.

3.2 Efficiency on Continuous Design Spaces

As is common practice in numerical design search, if the design space $\mathcal{C}$ is continuous, we consider optimal designs on $\mathcal{X}$, a set of grid points spread equidistantly in each variable. The finer the grid the more confident one can be about the optimality of the design derived numerically. However, computation time quickly increases with grid size, particularly in higher dimensions. For example, if $\mathcal{C}$ is three-dimensional, taking 100 points in each dimension results in 10^6 design points in $\mathcal{X}$. To find a balance between design performance and computation time, we investigate the relationship between grid size and design efficiency. Let ξ_p^* be a Φ_p-optimal design on $\mathcal{X}$ and ξ_p^c be a Φ_p-optimal design on $\mathcal{C}$ (ξ_p^c is not available in general). Define $\text{eff}_p^c(\xi_p^*) = \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_p^c})/\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_p^*})$ to be the Φ_p-efficiency of the design ξ_p^* on $\mathcal{C}$. The following theorem provides a lower bound for $\text{eff}_p^c(\xi_p^*)$.

Theorem 5. Let $\mathcal{X}$ be a grid on the continuous design space $\mathcal{C} \subset \mathcal{R}^r$, with grid points spread equidistantly in each variable with step size ε_j, $j = 1, \ldots, r$. Then $\text{eff}_p^c(\xi_p^*)$ is bounded from below by

$$\text{eff}_p^c(\xi_p^*) \geq \begin{cases} \exp\left(-\frac{1}{2v}\max\limits_{\mathbf{c}\in\mathcal{C}}\sum\limits_{j=1}^{r} B_j(\mathbf{c})\varepsilon_j\right), & \text{if } p = 0; \\ 1 - \frac{1}{2\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_p^*}(g))}\max\limits_{\mathbf{c}\in\mathcal{C}}\sum\limits_{j=1}^{r} B_j(\mathbf{c})\varepsilon_j, & \text{if } p > 0. \end{cases} \quad (3)$$

Here,

$$B_j(\mathbf{c}) = \begin{cases} n\left|\text{Tr}\left(\mathbf{M}\frac{\partial \mathbf{I}_\mathbf{c}}{\partial c_j}\right)\right|, & \text{if } p = 0; \\ n\left(\frac{1}{v}\right)^{\frac{1}{p}}\left(\text{Tr}\left(\mathbf{\Sigma}_{\xi_0+\xi_p^*}(g)\right)^p\right)^{\frac{1}{p}-1} & \text{if } p > 0; \\ \quad \times\left|\text{Tr}\left(\mathbf{M}\frac{\partial \mathbf{I}_\mathbf{c}}{\partial c_j}\right)\right|, \end{cases} \quad (4)$$

where c_i is the ith component of $\mathbf{c}$ and

$$\mathbf{M} = (\mathbf{I}_{\xi_0+\xi_p^*})^{-1}\left(\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right)^T\left(\mathbf{\Sigma}_{\xi_0+\xi_p^*}(g)\right)^{p-1}\left(\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right)(\mathbf{I}_{\xi_0+\xi_p^*})^{-1}.$$

Each term on the right-hand side of Equation (3) can be computed directly from programming except $\frac{\partial \mathbf{I}_\mathbf{c}}{\partial c_i}$, which we may have to compute by hand or by symbolic software, such as Maple or Mathematica. Computing $\max_{\mathbf{c}\in\mathcal{C}}\sum_{j=1}^{r} B_j(\mathbf{c})\varepsilon_j$ is challenging. If it cannot be found explicitly, we suggest finding a reasonably tight upper bound for this quantity, which can be substituted into Equation (3) to obtain another (slightly less tight) lower bound for the sought efficiency. To this end, consider $\mathcal{X}'$, a finer grid on $\mathcal{C}$, with grid points spread equidistantly in each variable with step size $\varepsilon_j' < \varepsilon_j$. For any $\mathbf{c} \in \mathcal{C}$, there exists a point $\mathbf{x}' \in \mathcal{X}'$, such that $|c_j - x_j'| \leq \varepsilon_j'/2$ for $j = 1, \ldots, r$, where, x_j' is the jth component of $\mathbf{x}'$. By the mean value theorem and the Cauchy-Schwarz inequality, we can show that

$$\left|\text{Tr}\left(\mathbf{M}\frac{\partial \mathbf{I}_\mathbf{c}}{\partial c_j}\right)\right| \leq \left|\text{Tr}\left(\mathbf{M}\frac{\partial \mathbf{I}_{\mathbf{x}'}}{\partial x_j'}\right)\right| + \frac{1}{2}|\mathbf{M}|\max_{\mathbf{c}'\in\mathcal{C}}\sqrt{\sum_{l=1}^{r}\left|\frac{\partial^2 \mathbf{I}_{\mathbf{c}'}}{\partial c_l'\partial c_j'}\right|^2}\sqrt{\sum_{l=1}^{r}(\varepsilon_l')^2}, \quad (5)$$

where $|.|$ is the L_2-norm. By Equations (4) and (5), with some rearranging of terms, we have

$$\max_{\mathbf{c}\in\mathcal{C}}\sum_{j=1}^{r} B_j(\mathbf{c})\varepsilon_j \leq \max_{\mathbf{x}'\in\mathcal{X}'}\sum_{j=1}^{r} B_j(\mathbf{x}')\varepsilon_j + \frac{1}{2}nD|\mathbf{M}|\sum_{j=1}^{r}\left(\varepsilon_j\max_{\mathbf{c}'\in\mathcal{C}}\sqrt{\sum_{l=1}^{r}\left|\frac{\partial^2 \mathbf{I}_{\mathbf{c}'}}{\partial c_l'\partial c_j'}\right|^2}\sqrt{\sum_{l=1}^{r}(\varepsilon_l')^2}\right). \quad (6)$$

Here, $D = 1$ if $p = 0$ and $D = (\frac{1}{v})^{\frac{1}{p}}(\text{Tr}(\mathbf{\Sigma}_{\xi_0+\xi_p^*}(g))^p)^{\frac{1}{p}-1}$ if $p > 0$. All terms on the right side of Equation (6) can be computed by SAS programming except for $\max_{\mathbf{c}'\in\mathcal{C}}\sqrt{\sum_{l=1}^{r}|\frac{\partial^2 \mathbf{I}_{\mathbf{c}'}}{\partial c_l\partial c_j'}|^2}$, which requires computation by hand or by symbolic software. We shall illustrate this method through an example in Section 4.

4. EXAMPLES

The most important feature of an algorithm is speed. Wu (1978) noted that "speed of approach, in the sense of computation time required is usually best dealt with empirically." In this section, we shall demonstrate, through several examples, that the algorithm is highly efficient. All coding was done in SAS IML, and computed on a Dell Laptop (2.2 GHz and 8 Gb RAM). The cut-off value for checking optimality was chosen to be $\epsilon_0 = 10^{-6}$. All derived optimal designs have been verified through Theorem 1.

The existing algorithms mainly focus on D-optimal design, when all parameters are of interest. Yu (2010) showed that the CA outperforms all existing algorithms by a large scale in this situation. We therefore compare the new algorithm only with the CA. We also assess the computation time of the new algorithm for different optimality criteria, different sets or functions of parameters of interest, and for multistage designs—scenarios other algorithms may not be directly applied to. Note that for multistage designs, in practice, we need to estimate the parameters, and then use the estimated parameters to select the design for the next stage. Since the aim here is to demonstrate the performance of the algorithm, we use the true parameters for illustration purposes.

For small step sizes ε_j, $j = 1, \ldots, r$, the number of design points in $\mathcal{X}$ increases rapidly, in particular for models with more than one explanatory variable, slowing down design search. We therefore also use a modified version of the new algorithm, and compare its performance with the original version. In the

Table 1. Computation time (seconds) for D-optimal designs for $\boldsymbol{\theta}$

	$N = 500$	$N = 1000$	$N = 5000$	$N = 10{,}000$
Cocktail	0.32	0.46	2.54	5.16
New algorithm	0.14	0.21	0.99	1.26
Modified algorithm	0.12	0.17	0.32	0.37

modified algorithm, we employ the multistage search strategy described by Stufken and Yang (2012): start with a coarse grid that is made increasingly finer in later stages; at each stage identify the best design based on the current grid. For the next stage, a finer grid is restricted to neighborhoods of the support points found at the current stage. The search continues until a specified accuracy for the design points is reached. The last step is to verify optimality through the equivalence theorem. From our experience, this strategy can further reduce the computation time. For illustration, the modified algorithm is applied to some selected problems below.

Example 1. Consider the nonlinear model

$$Y \sim \theta_1 e^{-\theta_2 x} + \theta_3 e^{-\theta_4 x} + N(0, \sigma^2),$$
$$\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4), \quad x \in [0, 3].$$

Let $(\theta_2, \theta_4) = (1, 2)$ and $\mathcal{X} = \{3i/N, i = 1, \ldots, N\}$. Yu (2010) found D-optimal designs for $\boldsymbol{\theta}$ (table 1 of Yu 2010). We code the CA in SAS IML and compare its computation time with the new algorithm for different grid sizes N in Table 1. We can see that, while the CA performs well, the new algorithm is about twice faster for moderate grid sizes, and even four times faster for finer grids. The modification reduces computation time even further.

The new algorithm is also highly efficient for different problems. We consider three different sets of parameters, $\boldsymbol{\theta}$, (θ_1, θ_3), and (θ_2, θ_4), and two different optimality criteria, D- and A-optimality. The computation time is less than 1 sec for almost all cases, even if there are 10,000 design points in $\mathcal{X}$ (see Table 2). Note that, although Theorem 3 does not imply that the new algorithm converges when partial parameters are of interest, it seems the new algorithm can still be applied to such problems.

Unlike the CA, the new algorithm can also be applied to multistage designs. Suppose we have an initial design $\xi_0 = \{(0, 0.25), (1, 0.25), (2, 0.25), (3, 0.25)\}$ with $n_0 = 40$. The problem is how to allocate the next 80 subjects. Again, we consider three sets of parameters, $\boldsymbol{\theta}$, (θ_1, θ_3), and (θ_2, θ_4), and D- and A-optimality. We can see from Table 3 that the performance is similar to that of locally optimal designs.

Table 2. Computation time (seconds) for different locally optimal designs

	D			A		
	$\boldsymbol{\theta}$	(θ_1, θ_3)	(θ_2, θ_4)	$\boldsymbol{\theta}$	(θ_1, θ_3)	(θ_2, θ_4)
$N = 500$	0.14	0.10	0.10	0.10	0.10	0.10
$N = 1000$	0.21	0.12	0.15	0.11	0.12	0.12
$N = 5000$	0.99	0.32	0.46	0.24	0.28	0.23
$N = 10{,}000$	1.26	0.54	0.85	0.45	0.42	0.45

Table 3. Computation time (seconds) for different multistage optimal designs

	D			A		
	$\boldsymbol{\theta}$	(θ_1, θ_3)	(θ_2, θ_4)	$\boldsymbol{\theta}$	(θ_1, θ_3)	(θ_2, θ_4)
$N = 500$	0.36	0.34	0.32	0.09	0.09	0.10
$N = 1000$	0.42	0.37	0.37	0.10	0.09	0.11
$N = 5000$	0.78	0.57	0.67	0.34	0.29	0.23
$N = 10{,}000$	1.27	0.78	0.98	0.54	0.57	0.40

Example 2. Consider the linear model

$$Y \sim \theta_1 + \theta_2 x_1 + \theta_3 x_1^2 + \theta_4 x_2 + \theta_5 x_1 x_2 + N(0, \sigma^2),$$
$$\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5), \tag{7}$$

where $(x_1, x_2) \in [-1, 1] \times [0, 1]$, and let $\mathcal{X} = \{(2i/s - 1, j/s), i = 0, 1, \ldots, s, j = 0, 1, \ldots, s\}$. Here, s is the number of grid points in each variable and the total number of points in $\mathcal{X}$ is $N = s^2$. Yu (2010) studied D-optimal design for $\boldsymbol{\theta}$ (table 4 of Yu 2010).

Table 4 shows that the new algorithm is again faster than the CA. As the grid size increases, this becomes more pronounced. For example, when $s = 200$ or 500, or equivalently $N = 200^2$ or $N = 500^2$, the new algorithm is about five times faster. The modification further reduces computation time, in particular for fine grids.

We also assessed the new algorithm on different problems; D- and A-optimality; all or just partial parameters of $\boldsymbol{\theta}$; locally optimal or multistage designs (tables not shown). The performance is similar, with computation times less than 1 sec for most cases, about 1.5 sec for grid size $N = 200^2$, and about 10 sec for grid size $N = 500^2$.

We next investigate how $\text{eff}_p^c(\xi_p^*)$, the Φ_p-efficiency of the design ξ_p^* on $\mathcal{C} = [-1, 1] \times [0, 1]$, changes with N. Suppose $\xi_0 = \{[(0.2, -1), 0.25], [(0.5, 0), 0.25], [(0.8, 1), 0.25], [(0.5, 0.5), 0.25]\}$ with $n_0 = 40$ is the initial design, and that a further 120 subjects are to be allocated. We compute lower bounds for $\text{eff}_p^c(\xi_p^*)$, according to Theorem 5, for a variety of problems.

For model (7), the information matrix of a single point $\mathbf{x} = (x_1, x_2)$, $\mathbf{I}_\mathbf{x} = f(\mathbf{x}) f(\mathbf{x})^T$, where $f(\mathbf{x}) = (1, x_1, x_1^2, x_2, x_1 x_2)^T$. Hence, $\frac{\partial \mathbf{I}(\mathbf{x})}{\partial x_1} = f_1(\mathbf{x}) f(\mathbf{x})^T + f(\mathbf{x}) f_1(\mathbf{x})^T$, where $f_1(\mathbf{x}) = (0, 1, 2x_1, 0, x_2)^T$, and $\frac{\partial \mathbf{I}(\mathbf{x})}{\partial x_2} = f_2(\mathbf{x}) f(\mathbf{x})^T + f(\mathbf{x}) f_2(\mathbf{x})^T$, where $f_2(\mathbf{x}) = (0, 0, 0, 1, x_1)^T$. With some algebra, we can show that

$$\left| \frac{\partial^2 \mathbf{I}(\mathbf{x})}{\partial^2 x_1} \right|^2 + \left| \frac{\partial^2 \mathbf{I}(\mathbf{x})}{\partial x_1 \partial x_2} \right|^2 \le 350 \quad \text{and}$$
$$\left| \frac{\partial^2 \mathbf{I}(\mathbf{x})}{\partial x_1 \partial x_2} \right|^2 + \left| \frac{\partial^2 \mathbf{I}(\mathbf{x})}{\partial^2 x_2} \right|^2 \le 78. \tag{8}$$

Table 4. Computation time (seconds) for D-optimal designs for $\boldsymbol{\theta}$

	$N = 20^2$	$N = 50^2$	$N = 100^2$	$N = 200^2$	$N = 500^2$
Cocktail	0.20	0.82	2.30	8.68	53.69
New algorithm	0.15	0.24	0.51	1.66	11.03
Modified algorithm	0.13	0.17	0.24	0.35	1.29

Table 5. Lower bounds for $\text{eff}_p^c(\xi_p^*)$ for different grid sizes N

	$\boldsymbol{\theta}$			$(\theta_2, \ldots, \theta_5)$		
	D	A	Φ_2	D	A	Φ_2
$N = 20^2$	0.872	0.680	0.662	0.843	0.676	0.662
$N = 50^2$	0.947	0.872	0.865	0.934	0.870	0.865
$N = 100^2$	0.973	0.936	0.932	0.966	0.935	0.932
$N = 200^2$	0.986	0.968	0.966	0.983	0.968	0.966
$N = 500^2$	0.995	0.987	0.986	0.993	0.987	0.986

For given s, ε_1 and ε_2 (defined in Theorem 5) are $\frac{2}{s}$ and $\frac{1}{s}$, respectively. Hence, $\sqrt{\sum_{j=1}^{2} \varepsilon_j^2} = \sqrt{5}/s$. We define the grid $\mathcal{X}'$, with points spread equidistantly in each variable with step sizes $2/3000$ and $1/3000$, respectively. We consider two different sets of parameters of interest, $\boldsymbol{\theta}$ and $(\theta_2, \ldots, \theta_5)$, and three different optimality criteria, D-, A-, and Φ_2-optimality. Applying Theorem 5 and using Equations (6) and (8), we obtain the lower bounds for $\text{eff}_p^c(\xi_p^*)$ for different grid sizes N in Table 5.

As N increases, the lower bounds for $\text{eff}_p^c(\xi_p^*)$ increase. For $N = 100^2$, the lower bounds are high (> 0.93), while the true efficiencies could be much higher than the lower bounds. For example, for $N = 20^2$, the A-optimal design for $\boldsymbol{\theta}$ gives an optimality value of 0.203818 while the corresponding optimality value for $N = 500^2$ is 0.203797, which implies that the A-optimal design for $N = 20^2$ is at least 98.7% efficient. However, to find this tighter bound we require the optimal design for $N = 500^2$ and its lower bound for efficiency.

Example 3. We consider a multinomial model with three different categories, that is, a response $\mathbf{Y} = (Y_1, Y_2, Y_3)^T$, with $Y_1 + Y_2 + Y_3 = 1$, at experimental condition $\mathbf{x}$ that has a multinomial distribution with parameters $\pi_1(\mathbf{x})$, $\pi_2(\mathbf{x})$, $1 - \pi_1(\mathbf{x}) - \pi_2(\mathbf{x})$, where

$$\pi_i(\mathbf{x}) = P(Y_i = 1|\mathbf{x}) = \frac{e^{g(\mathbf{x})^T\theta_i}}{1 + e^{g(\mathbf{x})^T\theta_1} + e^{g(\mathbf{x})^T\theta_2}}, i = 1, 2. \quad (9)$$

The vectors $g(\mathbf{x})$ usually hold lower-order monomials in the covariates $\mathbf{x} = (x_1, \ldots, x_r)^T$ and θ_1 and θ_2 are the corresponding coefficient vectors. The log-likelihood for a single observation and parameter vector $\boldsymbol{\theta} = (\theta_1^T, \theta_2^T)^T \in \mathbb{R}^k$ is then

$$l(\boldsymbol{\theta}; Y) = Y_1 \log \pi_1(\mathbf{x}) + Y_2 \log \pi_2(\mathbf{x}) + (1 - Y_1 - Y_2) \log(1 - \pi_1(\mathbf{x}) - \pi_2(\mathbf{x})),$$

and we obtain the information matrix at a single point $\mathbf{x}$ as

$$\mathbf{I}_\mathbf{x} = \left(\begin{array}{c|c} \pi_1(\mathbf{x})(1 - \pi_1(\mathbf{x}))\,\mathbf{J}(\mathbf{x}) & -\pi_1(\mathbf{x})\pi_2(\mathbf{x})\,\mathbf{J}(\mathbf{x}) \\ \hline -\pi_1(\mathbf{x})\pi_2(\mathbf{x})\,\mathbf{J}(\mathbf{x}) & \pi_2(\mathbf{x})(1 - \pi_2(\mathbf{x}))\,\mathbf{J}(\mathbf{x}) \end{array} \right),$$

where $\mathbf{J}(\mathbf{x}) = g(\mathbf{x})g^T(\mathbf{x})$. Note that the information matrix $\mathbf{I}_\mathbf{x}$ at a single point cannot be written in the form $\mathbf{I}_\mathbf{x} = f(\mathbf{x})f(\mathbf{x})^T$ for any vector function $f(\mathbf{x})$. Consequently, the CA cannot be applied to this example directly.

Multinomial logistic models are commonly used in toxicology experiments (see, e.g., Speybroeck et al. 2008, for an immunization experiment in cattle, with three categories, where $\mathbf{x}$ is the log-dose of the immunization treatment). To make this scenario more challenging for our algorithm, we add two further explanatory variables to the model, which corresponds to a situation where the vaccine is enhanced by two further substances to boost the immune reaction, or where the vaccine is composed of three different strains of the diluted parasite. We consider linear predictors, that is, $g(\mathbf{x}) = (1, \mathbf{x}^T)^T = (1, x_1, x_2, x_3)^T$ where $\mathbf{x} \in [0, 6]^3$, and parameter vectors $\boldsymbol{\theta}_1 = (\theta_{10}, \theta_{11}, \theta_{12}, \theta_{13})$ and $\boldsymbol{\theta}_2 = (\theta_{20}, \theta_{21}, \theta_{22}, \theta_{23})$.

Optimal designs for model (9) depend on the values of the unknown parameters (Zocchi and Atkinson 1999). We assume that $\boldsymbol{\theta}_1 = (1, 1, -1, 2)$ and $\boldsymbol{\theta}_2 = (-1, 2, 1, -1)$, and let the design space $\mathcal{X} = \{(6i/s, 6j/s, 6l/s), i, j, l = 0, 1, \ldots, s\}$. Due to the large size of $\mathcal{X}$, for this example we employ the modified algorithm only.

We consider (i) two different sets of parameters, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}' = (\theta_{11}, \theta_{12}, \theta_{13}, \theta_{21}, \theta_{22}, \theta_{23})$; (ii) both D- and A-optimality; and (iii) locally optimal design and multistage design with $\xi_0 = \{[(1, 3, 6), 0.25], [(4, 2, 1), 0.25], [(0, 1, 2), 0.25], [(2, 1, 0), 0.25]\}$ and $n_0 = 40$. Table 6 shows the computation times for different grid sizes.

Example 4. Dette, Melas, and Shpilev (2011) studied optimal designs for estimating the derivative of the expected response in nonlinear regression models. The optimal designs are determined numerically, based on some recursive formulas they derived. They considered following two models, (10) and (11), to demonstrate their methods:

$$Y \sim \theta_1 e^{\theta_2 x} + \theta_3 e^{\theta_4 x} + N(0, \sigma^2), \quad (10)$$

$$Y \sim \frac{\theta_1}{x + \theta_2 x} + \frac{\theta_3}{x + \theta_4 x} + N(0, \sigma^2), \quad (11)$$

with $x \in [0, 1]$. The function of parameters of interest, $g(\boldsymbol{\theta})$, is $\frac{\partial}{\partial x}(\theta_1 e^{\theta_2 x} + \theta_3 e^{\theta_4 x})$ for model (10) and $\frac{\partial}{\partial x}(\frac{\theta_1}{x+\theta_2 x} + \frac{\theta_3}{x+\theta_4 x})$ for

Table 6. Computation time (seconds) of the modified algorithm

	Locally optimal designs				Multistage optimal designs			
	$\boldsymbol{\theta}$		$\boldsymbol{\theta}'$		$\boldsymbol{\theta}$		$\boldsymbol{\theta}'$	
	D	A	D	A	D	A	D	A
$N = 10^3$	0.32	0.32	0.26	0.39	0.29	0.31	0.18	0.23
$N = 20^3$	0.62	1.07	1.15	1.71	0.74	1.73	0.65	1.34
$N = 50^3$	8.14	17.81	17.94	24.52	12.85	16.98	11.29	19.57
$N = 100^3$	54.38	86.92	101.13	169.57	71.73	68.14	71.09	114.64
$N = 200^3$	524.1	653.0	664.4	814.3	531.8	738.7	718.2	853.8

model (11). Since $g(\boldsymbol{\theta})$ is a scalar, this is a c-optimal design problem. Assuming $(\theta_1, \theta_2, \theta_3, \theta_4) = (1, 0.5, 1, 1)$, and interest is in $g(\boldsymbol{\theta})$ at the point $x = 0$ for both models, the optimal designs provided by Dette, Melas, and Shpilev (2011) are {(0, 0.3509), (0.3011, 0.4438), (0.7926, 0.1491), (1, 0.0562)} for model (10) and {(0, 0.3509), (0.0952, 0.4419), (0.4707, 0.1479), (1, 0.0597)} for model (11).

We apply the new algorithm to this problem. Either A- or D-optimality for $g(\boldsymbol{\theta})$ will yield the desired optimal designs. We use D-optimality here, and consider the grid $\mathcal{X} = \{i/10000, i = 0, 1, \ldots, 10000\}$. We obtain {(0, 0.3508), (0.3011, 0.4438), (0.7926, 0.1491), (1, 0.0563)} for model (10) and {(0, 0.3504), (0.0952, 0.4415), (0.4705, 0.1480), (1, 0.0601)} for model (11). The optimal designs provided by the new algorithm are not exactly the same as those found by Dette, Melas, and Shpilev (2011), which may be due to floating errors during the numerical computation. Our designs actually give slightly smaller optimal values if we do not round up to four decimal places. The computation time for deriving these designs was 0.42 sec for model (10) and 0.34 sec for model (11). We also tested the algorithm under different scenarios, such as different parameter values, different optimality criteria, and locally optimal or multistage designs. The computation times were all less than 1 sec.

5. DISCUSSION

While the importance of optimal/efficient designs in scientific studies cannot be disputed, their application in practice is not well established. The main reason is the lack of availability of efficient designs, caused by the lack of a general and efficient algorithm. The existing algorithms mainly focus on a specific optimality problem: locally D-optimal design for all parameters. We have demonstrated in several examples that the new algorithm outperforms the CA, which appears to be the best available algorithm so far for that specific problem. Moreover, the new algorithm can be applied to a much broader class of optimality problems: any set of differentiable functions of the parameters of interest; all Φ_p-optimality criteria with p being integer; locally optimal or multistage design. For all problems, the new algorithm performs efficiently; for most cases, we get instantaneous results. We believe this can greatly facilitate the application of optimal/efficient designs in practice.

Theorems 2 and 3 do not guarantee the convergence in some situations, for example, singular $\mathbf{I}_{\xi_0}$ (for multistage designs) or $\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}$ not being a full rank square matrix (for locally optimal designs). However, we experienced convergence in virtually all different situations. It may be worthwhile to study the theoretical properties for these cases. On the other hand, we feel the idea in this article can be extended to Bayesian optimal design, where numerical approaches are even more important. More research is certainly needed in this direction.

The coding of the new algorithm is more complicated than that of the existing algorithms. However, the main body of the code is the same for all models, with the only part requiring change is the form of the information matrix. The SAS IML codes for all examples in this article can be downloaded from *http://homepages.math.uic.edu/~minyang*. These codes can be easily modified for different optimality problems.

APPENDIX

Proof of Theorem 2. We only give the proof for $p > 0$. For $p = 0$, the proof is exactly the same with $\Phi_p(\boldsymbol{\Sigma}_{\xi_0+\xi}(g))$ replaced by $\log|\boldsymbol{\Sigma}_{\xi_0+\xi}(g)|$.

We begin the proof by establishing the convexity of $\Phi_p(\boldsymbol{\Sigma}_{\xi_0+\xi}(g))$ for $p \geq 0$.

Lemma 1. Suppose that $\frac{\partial g}{\partial \boldsymbol{\theta}^T}$ is a matrix of full row rank r. Then for any $0 \leq \epsilon \leq 1$, we have

$$\Phi_p(\boldsymbol{\Sigma}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}(g)) \leq (1-\epsilon)\Phi_p(\boldsymbol{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) + \epsilon\Phi_p(\boldsymbol{\Sigma}_{\xi_0+\xi^*}(g)).$$

Proof of Lemma 1. Since $\frac{\partial g}{\partial \boldsymbol{\theta}^T}$ is of full row rank, there exist a nonsingular matrix $\mathbf{A}$, such that $\frac{\partial g}{\partial \boldsymbol{\theta}} = (\mathbf{I_r}\ \mathbf{0})\mathbf{A}$, where $\mathbf{I}_r$ is an identity matrix and $\mathbf{0}$ is a zero-matrix with appropriate dimensions. Hence,

$$\begin{aligned}
&\boldsymbol{\Sigma}^{-1}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}(g) \\
&\quad= \left(\frac{\partial g}{\partial \boldsymbol{\theta}^T}\mathbf{I}^{-1}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}\left(\frac{\partial g}{\partial \boldsymbol{\theta}^T}\right)^T\right)^{-1} \\
&\quad= \left((\mathbf{I_r}\ \ \mathbf{0})((\mathbf{A}^T)^{-1}\mathbf{I}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}\mathbf{A}^{-1})^{-1}\begin{pmatrix}\mathbf{I_r}\\ \mathbf{0}^T\end{pmatrix}\right)^{-1}. \qquad \text{(A.1)}
\end{aligned}$$

By the definition of Schur complement (Pukelsheim 2006, sec. 3.11), $\boldsymbol{\Sigma}^{-1}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}(g)$ is the Schur complement of the first $r \times r$ principal submatrix of $(\mathbf{A}^T)^{-1}\mathbf{I}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}\mathbf{A}^{-1}$. Similarly, $\boldsymbol{\Sigma}^{-1}_{\xi_0+\xi_{S^{(t)}}}(g)$ and $\boldsymbol{\Sigma}^{-1}_{\xi_0+\xi^*}(g)$ are the Schur complements of the first $r \times r$ principal submatrix of $(\mathbf{A}^T)^{-1}\mathbf{I}_{\xi_0+\xi_{S^{(t)}}}\mathbf{A}^{-1}$ and $(\mathbf{A}^T)^{-1}\mathbf{I}_{\xi_0+\xi^*}\mathbf{A}^{-1}$, respectively. Note that

$$\begin{aligned}
&(\mathbf{A}^T)^{-1}\mathbf{I}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}\mathbf{A}^{-1} \\
&\quad= (1-\epsilon)(\mathbf{A}^T)^{-1}\mathbf{I}_{\xi_0+\xi_{S^{(t)}}}\mathbf{A}^{-1} + \epsilon(\mathbf{A}^T)^{-1}\mathbf{I}_{\xi_0+\xi^*}\mathbf{A}^{-1}. \qquad \text{(A.2)}
\end{aligned}$$

By concavity of the Schur complement (Pukelsheim 2006, sec. 3.11),

$$\boldsymbol{\Sigma}^{-1}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}(g) \geq (1-\epsilon)\boldsymbol{\Sigma}^{-1}_{\xi_0+\xi_{S^{(t)}}}(g) + \epsilon\boldsymbol{\Sigma}^{-1}_{\xi_0+\xi^*}(g). \qquad \text{(A.3)}$$

By Equation (A.3),

$$\begin{aligned}
&\Phi_p(\boldsymbol{\Sigma}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}(g)) \\
&\quad= \left(\frac{1}{v}\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}(g))^{-p}\right)^{1/p} \\
&\quad\leq \left(\frac{1}{v}\mathrm{Tr}((1-\epsilon)\boldsymbol{\Sigma}^{-1}_{\xi_0+\xi_{S^{(t)}}}(g) + \epsilon\boldsymbol{\Sigma}^{-1}_{\xi_0+\xi^*}(g))^{-p}\right)^{1/p} \\
&\quad\leq (1-\epsilon)\left(\frac{1}{v}\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}_{\xi_0+\xi_{S^{(t)}}}(g))^{-p}\right)^{1/p} + \epsilon\left(\frac{1}{v}\mathrm{Tr}(\boldsymbol{\Sigma}^{-1}_{\xi_0+\xi^*}(g))^{-p}\right)^{1/p} \\
&\quad= (1-\epsilon)\Phi_p(\boldsymbol{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) + \epsilon\Phi_p(\boldsymbol{\Sigma}_{\xi_0+\xi^*}(g)). \qquad \text{(A.4)}
\end{aligned}$$

The first and the second inequalities in Equation (A.4) follow from monotonicity and convexity of $\Psi_p(\mathbf{M})$, respectively (Fedorov and Hackl 1997, sec. 2.2), where $\Psi_p(\mathbf{M}) = (v^{-1}Tr(\mathbf{M}^{-p}))^{1/p}$. □

Now let Ξ be the set of all approximate designs ξ. Denote ξ^* as an optimal design that minimizes $\Phi_p(\boldsymbol{\Sigma}_{\xi_0+\xi}(g))$. Since $\mathbf{I}_{\xi_0}$ is nonsingular and $n_0 > 0$, for any $\xi \in \Xi$, we have

$$\Phi_p(\boldsymbol{\Sigma}_{\xi_0+\xi^*}(g)) \leq \Phi_p(\boldsymbol{\Sigma}_{\xi_0+\xi}(g)) \leq \Phi_p(\boldsymbol{\Sigma}_{\xi_0}(g)), \qquad \text{(A.5)}$$

where $\boldsymbol{\Sigma}_{\xi_0}(g) = \frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}(n_0\mathbf{I}_{\xi_0})^{-1}\{\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\}^T$. In addition, $\mathbf{I}_{\xi_0+\xi}$ is nonsingular regardless of ξ. Thus, $\Phi_p(\boldsymbol{\Sigma}_{\xi_0+(1-\alpha)\xi_1+\alpha\xi_2}(g))$ is infinitely differentiable with respect to α. Combining this fact with Equation (A.5), there exists $K < \infty$, such that

$$\sup\left\{\frac{\partial^2\Phi_p(\boldsymbol{\Sigma}_{\xi_0+(1-\alpha)\xi_1+\alpha\xi_2}(g))}{\partial\alpha^2} : \xi_1 \in \Xi, \xi_2 \in \Xi, \alpha \in [0,1]\right\} = K. \qquad \text{(A.6)}$$

The convergence of $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g))$ is obvious since it is a decreasing nonnegative function of t. We shall show that

$$\lim_{t\to\infty} \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) = \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi^*}(g)). \tag{A.7}$$

If Equation (A.7) does not hold, by the monotonicity of $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g))$, there exists $\delta > 0$, such that $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) - \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi^*}(g)) > \delta$ for all t. Utilizing the convexity of Φ_p, for any $0 \le \epsilon \le 1$, we have $\Phi_p(\mathbf{\Sigma}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}(g)) \le (1-\epsilon)\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) + \epsilon\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi^*}(g))$, which implies that

$$\begin{aligned}\lim_{\epsilon\downarrow 0} \frac{1}{\epsilon}(\Phi_p(\mathbf{\Sigma}_{\xi_0+(1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*}(g)) - \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)))& \\ \le \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi^*}(g)) - \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)).&\end{aligned} \tag{A.8}$$

The left hand side of Equation (A.8) is the first derivative of $\Phi_p(\mathbf{\Sigma}_{\xi_0+(1-\alpha)\xi_{S^{(t)}}+\alpha\xi^*}(g))$ with respect to α when $\alpha = 0$. By some standard matrix differentiation approach, using Equations (A.8), (1), and the definition of $\mathbf{x}_t^*$ in Equation (2), for all t, we have

$$d_p(\mathbf{x}_t^*, \xi_{S^{(t)}}) \ge \delta. \tag{A.9}$$

Consider a differently updated design $\xi_{S^{(t+1)}}(\alpha) = (1-\alpha)\xi_{S^{(t)}} \bigcup (x_t^*, \alpha)$, where $0 \le \alpha \le 1$. By the definition of $S^{(t+1)}$ in Equation (2), for any $\alpha \in [0, 1]$,

$$\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t+1)}}}(g)) \le \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t+1)}(\alpha)}}(g)). \tag{A.10}$$

Expanding into a Taylor series in α, and applying Equations (A.6) and (A.9), we can show that

$$\begin{aligned}&\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t+1)}(\alpha)}}(g)) \\ &= \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) - d_p(x_t^*, \xi_{S^{(t)}})\alpha \\ &\quad + \frac{1}{2}\alpha^2 \frac{\partial^2 \Phi_p(\mathbf{\Sigma}_{\xi_0+(1-\alpha)\xi_1+\alpha\xi_2})}{\partial\alpha^2}|_{\alpha=\alpha'} \\ &\le \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) - \delta\alpha + \frac{1}{2}K\alpha^2,\end{aligned} \tag{A.11}$$

where $\alpha' \in [0, \alpha]$. If $K > \delta$, let $\alpha = \frac{\delta}{K}$. By Equation (A.11), we have

$$\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t+1)}(\frac{\delta}{K})}}(g)) - \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) \le -\frac{\delta^2}{2K}. \tag{A.12}$$

By Equations (A.10) and (A.12), we have, for all $t \ge 0$,

$$\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t+1)}}}(g)) - \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) \le -\frac{\delta^2}{2K}. \tag{A.13}$$

Inequality (A.13) implies that $\lim_{t\to\infty} \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g)) = -\infty$, which contradicts the fact that $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_{S^{(t)}}}(g))$ is a nonnegative function. Similar arguments can be applied to the case when $K \le \delta$, in which we let $\alpha = 1$. □

Proof of Theorem 3. Again, we only give the proof for $p > 0$.

When $n_0 = 0$, there is no initial design ξ_0, that is, $\mathbf{\Sigma}_{\xi_0+\xi}(g) = \mathbf{\Sigma}_\xi(g)$, where

$$\mathbf{\Sigma}_\xi(g) = \frac{1}{n}\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\mathbf{I}_\xi^{-1}\left\{\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right\}^T. \tag{A.14}$$

Define $\Xi_1 = \{\xi : \Phi_p(\mathbf{\Sigma}_\xi(g)) \le \Phi_p(2\mathbf{\Sigma}_{\xi_{S^{(0)}}}(g))\}$. Clearly $\xi_{S^{(t)}} \in \Xi_1$ since $\Phi_p(\mathbf{\Sigma}_{\xi_{S^{(t)}}}(g))$ is a decreasing nonnegative function. Consider a differently updated design $\xi_{S^{(t+1)}}(\alpha) = (1-\alpha)\xi_{S^{(t)}} \bigcup (\mathbf{x}_t^*, \alpha)$. For any $\alpha \in [0, \frac{1}{2}]$, we have $\Phi_p(\mathbf{\Sigma}_{\xi_{S^{(t+1)}(\alpha)}}(g)) \le \Phi_p(2\mathbf{\Sigma}_{\xi_{S^{(t)}}}(g))$, which implies that $\xi_{S^{(t+1)}}(\alpha) \in \Xi_1$ for any $\alpha \in [0, \frac{1}{2}]$.

Since $\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}$ is a full rank square matrix, $\mathbf{I}_\xi$ is nonsingular for any $\xi \in \Xi_1$. Thus, the function $\Phi_p(\mathbf{\Sigma}_{(1-\alpha)\xi_1+\alpha\xi_2}(g))$ is infinitely differentiable with respect to α for any $\alpha \in [0, \frac{1}{2}]$. Combining this fact with Equation (A.14), there exists $M_1 < \infty$, such that

$$\sup\left\{\frac{\partial^2 \Phi_p(\mathbf{\Sigma}_{(1-\alpha)\xi_1+\alpha\xi_2}(g))}{\partial\alpha^2} : \xi_1 \in \Xi_1, \xi_2 \in \Xi, \alpha \in \left[0, \frac{1}{2}\right]\right\} = M_1.$$

The rest of the proof is the same as that of Theorem 2 with a minor but obvious modification. □

Proof of Theorem 4. By the standard theory of multivariate convex functions (see Kaplan 1999, sec. 1.9), it is sufficient to show that the Hessian of $\widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))$, $\frac{\partial^2 \widetilde{\Phi}_p(\mathbf{\Sigma}_{\xi_0+\xi}(g))}{\partial\boldsymbol{\omega}\partial\boldsymbol{\omega}^T}$, is a nonnegative definite matrix. Stufken and Yang (2012) proved this for $p = 0, 1$, for one-stage designs. Here we extend their results to nonnegative integers p, for multistage designs. Notice that the constraints imposed by Ω guarantee that the corresponding matrix $\mathbf{I}_{\xi_0+\xi}$ is a nonnegative definite matrix.

For simplification, we rewrite $\mathbf{\Sigma}_{\xi_0+\xi}(g)$ as $\mathbf{\Sigma}$ and $\mathbf{I}_{\xi_0+\xi}$ as $\mathbf{I}$. For $i = 1, \ldots, m-1$, define $\mathbf{I}^i = n(\mathbf{I}_{\mathbf{x}_i} - \mathbf{I}_{\mathbf{x}_m})$. Applying matrix differentiation, we have

$$\frac{\partial \mathbf{\Sigma}}{\partial \omega_i} = -\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\mathbf{I}^{-1}\mathbf{I}^i\mathbf{I}^{-1}\left\{\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right\}^T, \quad i = 1, \ldots, m-1, \tag{A.15}$$

$$\begin{aligned}\frac{\partial^2 \mathbf{\Sigma}}{\partial\omega_i\omega_j} = \frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}(\mathbf{I}^{-1}\mathbf{I}^j\mathbf{I}^{-1}\mathbf{I}^i\mathbf{I}^{-1} + \mathbf{I}^{-1}\mathbf{I}^i\mathbf{I}^{-1}\mathbf{I}^j\mathbf{I}^{-1})\left\{\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right\}^T,& \\ i, j = 1, \ldots, m-1.&\end{aligned} \tag{A.16}$$

Case (i): $p = 0$. Applying matrix differentiation, we obtain

$$\frac{\partial \log|\mathbf{\Sigma}|}{\partial\omega_i} = \mathrm{Tr}\left(\mathbf{\Sigma}^{-1}\frac{\partial \mathbf{\Sigma}}{\partial\omega_i}\right), \quad i = 1, \ldots, m-1, \tag{A.17}$$

$$\begin{aligned}\frac{\partial^2 \log|\mathbf{\Sigma}|}{\partial\omega_i\omega_j} = \mathrm{Tr}\left(\mathbf{\Sigma}^{-1}\frac{\partial^2 \mathbf{\Sigma}}{\partial\omega_i\omega_j} - \mathbf{\Sigma}^{-1}\frac{\partial \mathbf{\Sigma}}{\partial\omega_j}\mathbf{\Sigma}^{-1}\frac{\partial \mathbf{\Sigma}}{\partial\omega_i}\right),& \\ i, j = 1, \ldots, m-1.&\end{aligned} \tag{A.18}$$

Using Equations (A.15) and (A.16), with some matrix algebra, we can show that

$$\begin{aligned}\frac{\partial^2 \log|\mathbf{\Sigma}|}{\partial\omega_i\omega_j} = \mathrm{Tr}&\left(\mathbf{\Sigma}^{-\frac{1}{2}}\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\mathbf{I}^{-1}\mathbf{I}^j\mathbf{I}^{-\frac{1}{2}}\mathbf{I}^{-\frac{1}{2}}\mathbf{I}^i\mathbf{I}^{-1}\left\{\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right\}^T\mathbf{\Sigma}^{-\frac{1}{2}}\right) \\ &+ \mathrm{Tr}\left(\mathbf{\Sigma}^{-\frac{1}{2}}\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\mathbf{I}^{-1}\mathbf{I}^j\mathbf{I}^{-\frac{1}{2}}P^\perp\left[\mathbf{I}^{-\frac{1}{2}}\left\{\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right\}^T\right]\right. \\ &\left.\times\ \mathbf{I}^{-\frac{1}{2}}\mathbf{I}^i\mathbf{I}^{-1}\left\{\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right\}^T\mathbf{\Sigma}^{-\frac{1}{2}}\right),\end{aligned}$$

where $P^\perp(\mathbf{X}) = \mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^-\mathbf{X}^T$ denotes the orthogonal projection matrix onto the orthogonal complement of the column space of $\mathbf{X}$. Thus, the Hessian matrix of $\log|\mathbf{\Sigma}|$ is nonnegative definite by Proposition 1 by Stufken and Yang (2012).

Case (ii): $p > 0$. Applying matrix differentiation, we have

$$\frac{\partial \mathrm{Tr}(\mathbf{\Sigma})^p}{\partial\omega_i} = p\mathrm{Tr}\left(\mathbf{\Sigma}^{p-1}\frac{\partial \mathbf{\Sigma}}{\partial\omega_i}\right), \quad i = 1, \ldots, m-1, \tag{A.19}$$

$$\begin{aligned}\frac{\partial^2 \mathrm{Tr}(\mathbf{\Sigma})^p}{\partial\omega_i\omega_j} = p\mathrm{Tr}\left(\mathbf{\Sigma}^{p-1}\frac{\partial^2 \mathbf{\Sigma}}{\partial\omega_i\omega_j} + \sum_{l=0}^{p-2}\mathbf{\Sigma}^l\frac{\partial \mathbf{\Sigma}}{\partial\omega_j}\mathbf{\Sigma}^{p-2-l}\frac{\partial \mathbf{\Sigma}}{\partial\omega_i}\right),& \\ i, j = 1, \ldots, m-1.&\end{aligned} \tag{A.20}$$

Note that when $p = 1$, the second term on the right-hand side of Equation (A.20) vanishes.

Using Equation (A.16), with some algebra, it can be shown that

$$\begin{aligned}&\mathrm{Tr}\left(\mathbf{\Sigma}^{p-1}\frac{\partial^2 \mathbf{\Sigma}}{\partial\omega_i\omega_j}\right) \\ &= 2\mathrm{Tr}\left(\mathbf{\Sigma}^{\frac{p-1}{2}}\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\mathbf{I}^{-1}\mathbf{I}^j\mathbf{I}^{-\frac{1}{2}}\mathbf{I}^{-\frac{1}{2}}\mathbf{I}^i\mathbf{I}^{-1}\left\{\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right\}^T\mathbf{\Sigma}^{\frac{p-1}{2}}\right),\end{aligned} \tag{A.21}$$

$$\begin{aligned}&\mathrm{Tr}\left(\mathbf{\Sigma}^l\frac{\partial \mathbf{\Sigma}}{\partial\omega_j}\mathbf{\Sigma}^{p-2-l}\frac{\partial \mathbf{\Sigma}}{\partial\omega_i}\right) \\ &= \mathrm{Tr}\left(\mathbf{\Sigma}^{\frac{l}{2}}\frac{\partial \mathbf{\Sigma}}{\partial\omega_j}\mathbf{\Sigma}^{\frac{p-2-l}{2}}\mathbf{\Sigma}^{\frac{p-2-l}{2}}\frac{\partial \mathbf{\Sigma}}{\partial\omega_i}\mathbf{\Sigma}^{\frac{l}{2}}\right).\end{aligned} \tag{A.22}$$

By Equations (A.20), (A.21), and (A.22), applying Proposition 1 by Stufken and Yang (2012), we conclude that the Hessian matrix of $Tr(\mathbf{\Sigma})^p$ is nonnegative definite. □

Proof of Theorem 5. Let ξ_p^c be a Φ_p-optimal design on $\mathcal{C}$. By the same argument as in the proof of Theorem 2, we have $\Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_p^*}(g)) - \Phi_p(\mathbf{\Sigma}_{\xi_0+\xi_p^c}(g)) \le \max_{\mathbf{c}\in\mathcal{C}} d_p(\mathbf{c}, \xi_p^*)$.

By the construction of $\mathcal{X}$, for any $\mathbf{c} \in \mathcal{C}$, there exists $\mathbf{x}^c \in \mathcal{X}$, such that $|c_j - x_j^c| \le \varepsilon_j/2$ for $j = 1, \ldots, r$, where c_j and x_j^c are the jth variable of $\mathbf{c}$ and $\mathbf{x}^c$, respectively. From the mean value theorem, there exists a scalar $\alpha \in (0, 1)$ such that

$$\begin{aligned} d_p(\mathbf{c}, \xi_p^*) &= d_p(\mathbf{x}^c, \xi_p^*) + \nabla d_p((1-\alpha)\mathbf{c} + \alpha\mathbf{x}^c, \xi_p^*) \cdot (\mathbf{c} - \mathbf{x}^c) \\ &\le \sum_{j=1}^{r} |\nabla_j d_p((1-\alpha)\mathbf{c} + \alpha\mathbf{x}^c, \xi_p^*)|\varepsilon_j/2, \end{aligned} \tag{A.23}$$

where $\cdot$ denotes the Euclidean inner product, ∇ denotes the gradient, and ∇_j denotes its jth element. The inequality in Equation (A.23) follows from $\max_{x\in\mathcal{X}} d_p(\mathbf{x}, \xi_p^*) = 0$. From the definition of d_p in Equation (1), it is straightforward to show that $|\nabla_j d_p((1-\alpha)\mathbf{c} + \alpha\mathbf{x}^c, \xi_p^*)| = B_j((1-\alpha)\mathbf{c} + \alpha\mathbf{x}^c)$, where B_j is defined in Equation (4). From the definition of efficiency, the conclusion follows. □

[Received February 2012. Revised April 2013.]

REFERENCES

Böhning, D. (1986), "A Vertex-Exchange-Method in *D*-Optimal Design Theory," *Metrika*, 33, 337–347. [1411]

Chernoff, H. (1953), "Locally Optimal Designs for Estimating Parameters," *Annals of Mathematical Statistics*, 24, 586–602. [1411]

——— (1999), "Gustav Elfving's Impact on Experimental Design," *Statistical Science*, 14, 201–205. [1411]

Covey Crump, P. A. K., and Silvey, S. D. (1970), "Optimal Regression Designs With Previous Observations," *Biometrika*, 57, 551–566. [1412]

Dette, H., and Melas, V. B. (2011), "A Note on the de la Garza Phenomenon for Locally Optimal Designs," *The Annals of Statistics*, 39, 1266–1281. [1411,1414]

Dette, H., Melas, V. B., and Shpilev, P. (2011), "Optimal Designs for Estimating the Derivative in Nonlinear Regression," *Statistica Sinica*, 21, 1557–1570. [1417,1418]

Dragalin, V., Hsuan, F., and Padmanabhan, S. K. (2007), "Adaptive Designs for Dose-Finding Studies Based on Sigmoid Emax Model," *Journal of Biopharmaceutical Statistics*, 17, 1051–1070. [1411]

Fedorov, V. V. (1972), *Theory of Optimal Experiments* (trans. and ed. W. J. Studden and E. M. Klimko), New York: Academic Press. [1411]

Fedorov, V. V., and Hackl, P. (1997), *Model-Oriented Design of Experiments*, New York: Springer. [1418]

Harman, R., and Pronzato, L. (2007), "Improvements on Removing Non-Optimal Support Points in *D*-Optimum Design Algorithms," *Statistics & Probability Letters*, 77, 90–94. [1411]

Hettich, R. (1983), "A Review of Numerical Methods for Semi-Infinite Optimization," *Lecture Notes in Economics and Mathematical Systems*, 215, 158–178. [1411]

Isaacson, E., and Keller, H. B. (1966), *Analysis of Numerical Methods*, New York: Wiley. [1414]

Kaplan, W. (1999), *Maxima and Minima With Applications*, New York: Wiley. [1414,1419]

Kiefer, J. (1974), "General Equivalence Theory for Optimum Designs (Approximate Theory) Extremum Problems," *The Annals of Statistics*, 2, 849–879. [1413]

Martín-Martín, R., Rodríguez-Aragón, L., and Torsney, B. (2012), "Multiplicative Algorithm for Computing *D*-Optimum Designs for pVT Measurements," *Chemometrics and Intelligent Laboratory Systems*, 111, 20–27. [1411]

Pukelsheim, F. (2006), *Optimal Design of Experiments*, Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM). [1418]

Pukelsheim, F., and Torsney, B. (1991), "Optimal Weights for Experimental Designs on Linearly Independent Support Points," *The Annals of Statistics*, 19, 1614–1625. [1413]

Silvey, S. D. (1980), *Optimal Design: An Introduction to the Theory for Parameter Estimation*, London: Chapman & Hall. [1412]

Silvey, S. D., Titterington, D. M., and Torsney, B. (1978), "An Algorithm for Optimal Designs on a Finite Design Space," *Communications in Statistics—Theory and Methods*, 14, 1379–1389. [1411]

Speybroeck, N., Marcotty, T., Aerts, M., Dolan, T., Williams, B., Lauer, J., Molenberghs, G., Burzykowski, T., Mulumba, M., and Berkvens, D. (2008), "Titrating Theileria Parva: Single Stocks Against Combination of Stocks," *Experimental Parasitology*, 118, 522–530. [1417]

Stufken, J., and Yang, M. (2012), "On Locally Optimal Designs for Generalized Linear Models With Group Effects," *Statistica Sinica*, 22, 1765–1786. [1416,1419]

Torsney, B. (1981), "Algorithms for a Constrained Optimization Problem With Applications in Statistics and Optimal Design," unpublished Ph.D. dissertation, University of Glasgow. [1411]

Torsney, B., and Martín-Martín, R. (2009), "Multiplicative Algorithms for Computing Optimum Designs," *Journal of Statistical Planning and Inference*, 139, 3947–3961. [1411]

Wu, C. F. J. (1978), "Some Iterative Procedures for Generating Nonsingular Optimal Designs," *Communications in Statistics—Theory and Methods*, 14, 1399–1412. [1415]

Wynn, H. P. (1970), "The Sequential Generation of *D*-Optimal Experimental Designs," *The Annals of Mathematical Statistics*, 41, 1655–1664. [1411]

Yang, M. (2010), "On the de la Garza Phenomenon," *The Annals of Statistics*, 38, 2499–2524. [1411,1414]

Yang, M., and Stufken, J. (2009), "Support Points of Locally Optimal Designs for Nonlinear Models With Two Parameters," *The Annals of Statistics*, 37, 518–541. [1411]

——— (2012), "Identifying Locally Optimal Designs for Nonlinear Models: A Simple Extension With Profound Consequences," *The Annals of Statistics*, 40, 1665–1681. [1414]

Yu, Y. (2010), "Monotonic Convergence of a General Algorithm for Computing Optimal Designs," *The Annals of Statistics*, 38, 1593–1606. [1415,1416]

——— (2011), "*D*-optimal Designs via a Cocktail Algorithm," *Statistics and Computing*, 21, 475–481. [1412]

Zocchi, S. S., and Atkinson, A. C. (1999), "Optimum Experimental Designs for Multinomial Logistic Models," *Biometrics*, 55, 437–444. [1417]