Noname manuscript No.
(will be inserted by the editor)

Optimal Minimax Designs via Particle Swarm Optimization Methods

Ray-Bing Chen · Shin-Perng Chang · Weichung Wang · Heng-Chih Tung · Weng Kee Wong

the date of receipt and acceptance should be inserted later

Abstract Particle swarm optimization (PSO) techniques are widely used in applied fields to solve challenging optimization problems but they do not seem to have made an impact in mainstream statistical applications hitherto. PSO methods are popular because they are easy to implement and use, and seem increasingly capable of solving complicated problems without requiring any assumption on the objective function to be optimized. We modify PSO techniques to find optimal minimax designs, which have been notoriously challenging to find to date even for linear models, and show that the PSO methods can readily generate a variety of optimal minimax designs, including optimal maximin designs in a novel and interesting way.

Keywords Continuous optimal design · equivalence theorem · Fisher information matrix · minimax optimality criteria · regression model

1 Introduction

Particle Swarm Optimization (PSO) is a population based stochastic optimization method inspired by social behavior of bird flocking or fish schooling and proposed by Eberhart and Kennedy (1995). In the last decade or so, PSO has singularly generated considerable interest in optimization circles as evident by its ever increasing applications in various disciplines. The importance and popularity of PSO can also be seen in the existence of many websites that provide PSO tutorials and PSO codes, track PSO development and applications in different fields. Some exemplary websites on PSO are `http://www.swarmintelligence.org/index.php`, `http://www.particleswarm.info/` and `http://www.cis.syr.edu/~mohan/pso/`. Currently, there are at least 3 journals that have a focus theme on swarm intelligence and applications with a few more having an emphasis on the more general

Ray-Bing Chen
Department of Statistics, National Cheng-Kung University, Tainan 70101

Shin-Perng Chang
Department of Network System, Toko University, Puzih, Chiayi 61363, Taiwan

Weichung Wang
Department of Mathematics, National Taiwan University, Taipei 10617, Taiwan

Heng-Chih Tung
Department of Statistics, National Cheng-Kung University, Tainan 70101, Taiwan

Weng Kee Wong
Department of Biostatistics, Fielding School of Public Health, UCLA, Los Angeles, CA 90095-1772, U.S.A.

class of nature-inspired metaheuristic algorithms, of which PSO is a member. Nature-inspired metaheuristic algorithms have been rising in popularity in the optimization literature in the last 2 decades and in the last decade have dominated the optimization world compared with traditional mathematical optimization tools (Whitacre, 2011a,b). Of particular note is Yang (2010), who saw a need to come up a second edition of his book on nature-inspired metaheuristic algorithms published less than 2 years earlier. This shows just how dynamic and rapidly expanding the field is. Clerc (2006) seems to be the first book devoted entirely to PSO and an updated overview of PSO methodology is available in Poli et al. (2007).

Interestingly, PSO has yet to make an impact in the statistical literature. We believe PSO methodology can be potentially useful to solve many statistical problems because ideas behind PSO are very simple and general yet requiring minimal or no assumption on the function to be optimized. Our aim is to show that PSO methodology is effective in finding many types of optimal designs, including optimal minimax designs, which are notoriously difficult to find and study. This is because the design criterion is non-differentiable and there is no effective algorithm for finding them to date, even for linear models. Specifically, we demonstrate that PSO can readily generate different types of optimal minimax designs for linear and nonlinear models that agree with the few published results in the literature.

PSO is a stochastically iterative procedure to optimize a function. The key advantages of this approach are that the PSO is fast and flexible, there are few tuning parameters required of the algorithm and the PSO codes can be generically written down easily to find optimal designs for a regression model. For more complicated problems, such as minimax design problems, the code will have to be modified appropriately. Generally, only the optimality criterion and the information matrix in the codes have to be changed to find optimal design for another problem. We discuss this further in the exemplary pseudo MATLAB codes that we provide in Section 4 to generate the optimal designs.

In the next section, we provide the background. In Section 3, we demonstrate that the PSO method can efficiently generate different types of optimal minimax designs for linear and nonlinear models. The PSO method usually takes a few seconds of CPU time to find the locally optimal designs, and sometimes even so for optimal minimax designs. In Section 4, we provide computational and implementation details for our proposed PSO-based procedure and Section 6. We also show PSO methodology can be modified to find standardized maximin optimal designs, which first require locally optimal design for each value of the parameter to be be available. As illustrative examples, we construct such designs for enzyme kinetic models in Section 5 before we close with a discussion.

2 Background

We focus on continuous designs that treat all designs as probability measures on the given design space X. This approach was proposed by Kiefer and his collection of voluminous work in this area is now documented in a single collection (Kiefer, 1985). If a continuous design takes p_i proportion of the total observations at $x_i \in X, i = 1, 2, \ldots, k$, we denote it by ξ with $p_1 + p_2 + \cdots + p_k = 1$. Given a fixed sample size N, we implement ξ by taking roughly Np_i observations at $x_i, i = 1, 2, .., k$ subject to $Np_1 + Np_2 + \cdots + Np_k = N$. As Kiefer had shown, one can round each of the Np_i's to the nearest integer so that they sum to N without losing too much efficiency if the sample size is large. The proportion p_i is sometimes called the weight of the design at x_i. Continuous designs are practical to work with, along with many other advantages widely documented in design monographs, such as Fedorov (1972), Silvey (1980), Pázman (1986), Atkinson et al. (2007) and in Kiefer (1985).

Our setup assumes we have a statistical model defined on given compact design region X. The mean of the univariate response is modeled by a known function $g(x, \theta)$ apart from the values of the vector of parameters θ. We assume errors are normally and independently distributed, all with means zero and possibly unequal variances. The mean function $g(x, \theta)$ can be a linear or nonlinear function of θ and the set of independent variables x. Following convention, the worth of the design ξ is measured by its Fisher information matrix defined to be the negative of the expectation of the matrix of second derivatives of the log-likelihood function. For example, consider the popular Michaelis-Menten model in the biological sciences given by

$$y = g(x, \theta) + \varepsilon = \frac{ax}{b + x} + \varepsilon, \quad x > 0,$$

where $a > 0$ denotes the maximal response possible and $b > 0$ is the value of x for which there is a half-maximal response. In practice, the design space is truncated to $X = [0, c]$ where c is a sufficiently large user-selected constant. If $\theta^\top = (a, b)$ and the errors ε are normally and independently distributed with means 0 and constant variance, the Fisher information matrix for a given design ξ is

$$I(\theta, \xi) = \int \frac{\partial g(x, \theta)}{\partial \theta} \frac{\partial g(x, \theta)}{\partial \theta^T} \xi(dx) = \int (\frac{ax}{b + x})^2 \begin{pmatrix} \frac{1}{a^2} & -\frac{1}{a(b+x)} \\ -\frac{1}{a(b+x)} & \frac{1}{(b+x)^2} \end{pmatrix} \xi(dx).$$

For nonlinear models, such as the Michaelis-Menten model, the information matrix depends on the model parameters. For linear models, the information matrix does not depend on the model parameters and we will denote it simply by $I(\xi)$.

Following convention, the optimality criterion is formulated as a convex function of the design and the optimal design is found by minimizing the criterion over all designs on the design space X. A common criterion is D-optimality where we want to find a design to minimize $\log |I(\theta, \xi)^{-1}|$ over all designs ξ on X. Because this criterion contains θ, a nominal value or best guess is needed for θ before the function is minimized. The resulting D-optimal design depends on the nominal value and so it is called locally D-optimal. Further, the criterion is a convex function in ξ and this means that a standard directional derivative argument can be applied to produce an equivalence theorem which checks whether a given design is D-optimal among all designs on X. Details are available in the above cited design monographs.

Optimal minimax design arises naturally when we wish to have protection against the worst case scenario. For example if the vector of model parameters is θ and Θ a user-selected set of plausible values for θ, one may want to implement an optimal minimax design ξ^* defined by

$$\xi^* = \arg\min_{\xi} \max_{\theta \in \Theta} \log |I^{-1}(\theta, \xi)|,$$

where the minimization is over all designs on X. The optimal design provides some global protection against the worst case scenario by minimizing the maximal inefficiencies of the parameter estimates. Clearly, when Θ is a singleton set, the optimal minimax design is the same as the locally optimal design. A common application of the minimax design criterion is in a dose response study where the goal is to find a extrapolation optimal design that provides the best inference on the mean responses over a known interval Z outside the dose interval X. If we have a heteroscedastic linear model with mean function $g(x)$ and $\lambda(x)$ is the assumed reciprocal variance of the response at dose x, the design ξ^* that provides the best inference for the mean response at the given dose $z \in Z$ is the one that minimizes

$$v(z, \xi) = g^T(z) I(\xi)^{-1} g(z)$$

among all designs ξ on X and

$$I(\xi) = \int \lambda(x) g(x) g^T(x) \xi(dx). \tag{1}$$

However if we know there are several dose levels of interest and they are all in some pre-determined compact set Z, one may seek a design to minimize the maximal variance of the fitted responses on Z. Such a design criterion is also convex and one can use the following equivalence theorem to check whether a design is minimax optimal for extrapolation on Z: ξ^* is minimax-optimal if and only if there exists a probability measure μ^* on $A(\xi^*)$ such that for all x in X,

$$c(x, \mu^*, \xi^*) = \int_{A(\xi^*)} \lambda(x) r(x, u, \xi^*) \mu^*(du) - v(u, \xi^*) \le 0$$

with equality at the support points of ξ^*. Here, $A(\xi) = \{u \in Z | v(u, \xi) = \max_{z \in Z} v(z, \xi)\}$ and $r(x, u, \xi) = (g^T(x) M(\xi)^{-1} g(u))^2$. If X is one or two-dimensional, one may visually inspect the plot of $c(x, \mu^*, \xi^*)$

versus values of $x \in X$ to confirm the optimality of ξ^*. In what is to follow, we display such plots to verify the optimality of a design without reporting the measure μ^*. A formal proof of this equivalence theorem can be found in Berger et al. (2000) and further details on the optimal minimax design problems are available in Wong (1992) and, Wong and Cook (1993) with further examples in King and Wong (1998, 2000). Extensions to nonlinear models are straightforward if one assumes the mean response can be adequately approximated by a linear model via a first order Taylor's expansion.

There are 3 points worth noting: (i) when Z is a singleton set that does not belong to X, the probability measure μ^* is necessarily degenerate at Z and the resulting equivalence theorem reduces to one for checking whether a design is c-optimal, see Fedorov (1972) or Silvey (1980); (ii) equivalence theorems for minimax optimality criteria all have a form similar to the one shown above and they are more complicated because we need to work with the subgradient μ^*. Finding the subgradient requires another set of optimization procedure which usually is more tricky to handle and this in part explains why optimal minimax designs are much harder to find than optimal designs under a differentiable criterion, and (iii) under the setup here, the convex design criterion allows us to derive a lower bound on the efficiency of any design (Pázman, 1986). This implies that one can always assess how good a design is by providing its efficiency lower bound (without knowing the optimal design).

3 PSO-generated Optimal Minimax Designs

Optimal minimax designs are notoriously difficult to find and we know of no algorithm to date that is guaranteed to find such optimal designs. Even for linear polynomial models with a few factors, recent papers acknowledged the difficulty of finding optimal minimax designs; see Rodriguez et al. (2010) and Johnson et al. (2011), where they considered finding a G-optimal design to minimize the maximal variance of the fitted response across the design space. Optimal minimax designs for nonlinear models can be challenging even when there are just two parameters in the model. Earlier attempts to solve such minimax problems have to impose constraints to simplify the optimization problem. For example, Sitter (1992) found minimax D-optimal designs for the two-parameter logistic model among designs that allocated equal number of observations at equally space points placed symmetrically about the location parameter. Similarly, Noubiap and Seidel (2000) found optimal minimax designs numerically among symmetric and balanced designs after noting that "by restricting the set of regarded designs in a suitable way, the minimax problem becomes numerically tractable in principle; nevertheless it is still a two-level problem requiring nested global optimization." In the same paper on p.152, the authors remarked that "Unfortunately, the minimax procedure is, in general, numerically intractable".

We are therefore naturally interested to investigate whether PSO methodology provides an effective way to find optimal minimax designs. Our examples in this section are confined to the scattered few optimal minimax designs reported in the literature, either numerically or analytically. The hope is that all optimal designs found by PSO agree with results in the literature and this would then suggest that the algorithm should also work well for problems whose optimal minimax designs are unknown. Of course, we can also confirm the optimality of the design found by the PSO method using an equivalence theorem. Example 3 below is one such instance.

We selectively present three examples and briefly a fourth with two independent variables out of many successes we have had with PSO for finding different types of optimal minimax designs. One of the examples has a binary response and the rest have continuous responses. The first example seeks to find a locally E-optimal design that minimizes the maximum eigenvalue of the inverse of the Fisher information matrix. Example 2 seeks a best design for estimating parameters in a two-parameter logistic model when we have apriori a range of plausible values for each of the two parameters. The desired design sought is the one that maximizes the smallest determinant of the information matrix over all nominal values of the two parameters in the plausible region. Equivalently, this is the optimal minimax design that minimizes the maximum determinant of the inverse of the information matrix where the maximum is taken over all nominal values in the plausible region for the parameters. The numerically optimal minimax design for Example 2 was found by repeated guess work followed by confirmation with the equivalence theorem in King and Wong (2000) with the aid of Mathematica. We will compare their designs with our PSO-generated designs. The third example concerns a heteroscedastic quadratic model

Table 1 Locally E-optimal designs for the Michaelis-Menten model with $[0, \tilde{x}] = [0, 200]$.

a	b	ξ_{PSO}		E-optimal designs	
100	150	46.520 (0.6925)	200 (0.3075)	45.510 (0.6927)	200 (0.3073)
100	100	38.152 (0.6770)	200 (0.3230)	38.150 (0.6769)	200 (0.3231)
100	50	24.783 (0.6171)	200 (0.3829)	24.780 (0.6171)	200 (0.3829)
100	10	6.516 (0.2600)	200 (0.7400)	6.515 (0.2600)	200 (0.7400)
100	1	0.701 (0.0222)	200 (0.9778)	0.701 (0.0220)	200 (0.9778)
10	150	46.497 (0.7071)	200 (0.2929)	46.510 (0.7070)	200 (0.2931)
10	100	38.142 (0.7068)	200 (0.2932)	38.150 (0.7068)	200 (0.2933)
10	50	24.778 (0.7058)	200 (0.2942)	24.780 (0.7058)	200 (0.2942)
10	10	6.515 (0.6837)	200 (0.3163)	6.515 (0.6838)	200 (0.3162)
10	1	0.701 (0.1882)	200 (0.8118)	0.701 (0.1881)	200 (0.8119)

with a known efficiency function and we want to find a design to minimize the maximum variance of the fitted responses across a user-specified interval. The optimal minimax designs are unknown for this example and we will check the optimality of the PSO-generated design using an equivalence theorem.

The key tuning parameters in the PSO method are (i) flock size, i.e. number of particles (designs) to use in the search, (ii) the number of common support points the particles have, and (iii) the number of iterations allowed in the search process. Since there are two optimization problems involved, we need to specify these numbers for each optimization problem and we prefix them with either outer or inner, with the outer designation referring to the first optimization problem that finds the maximal value and the inner designation for find the design that minimizes the maximal value. Unless mentioned otherwise, we use the same values for these outer and inner numbers and omit the outer or inner reference. We use default values for all other tuning parameters in the PSO codes that we programmed in MATLAB version R2010b 64bit. Section 4 provides information on these default values. All CPU computing times (in seconds) were from a Intel Core2 6300 computer with 5 GB RAM and operating system Ubuntu 64bit Linux with kernel 2.6.35-30.

Before we present our modified PSO method called Nested PSO in Section 4, we present four examples, with a bit more details for the first example.

3.1 Example 1: E-optimal designs for the Michaelis-Menten model.

The Michaelis-Menten model is one of the simplest and most widely used model in the biological sciences. Dette and Wong (1999) used a geometric argument based on the celebrated Elfving's theorem and constructed locally E-optimal designs for the model with two parameters $\theta^\top = (a, b)$. Such optimal designs are useful for making inference on θ by making the area of the confidence ellipsoid small in terms of minimizing the length of the longest principal axis. This is achieved by minimizing the larger of the two eigenvalues of the inverse of the information matrix over all designs on X. For a given θ, they showed that if the known design space is $X = [0, \tilde{x}]$ and $\tilde{z} = \tilde{x}/(b + \tilde{x})$, the locally E-optimal design is supported at $\tilde{x}$ and $\{(\sqrt{2} - 1)b\tilde{x}\}/\{2 - \sqrt{2})\tilde{x} + b\}$ and the weight at the latter support point is

$$w = \frac{\sqrt{2}(a/b)^2(1 - \tilde{z})\{\sqrt{2} - (4 - 2\sqrt{2})\tilde{z}\}}{2 + (a/b)^2\{\sqrt{2} - (4 - 2\sqrt{2})\tilde{z}\}^2}.$$

We use the Nested PSO procedure to search for the locally 2-point E-optimal design using 128 particles and 100 iterations. Selected optimal minimax designs are shown in Table 1 along with the theoretical optimal designs reported in Dette and Wong (1999). All PSO-generated designs are close to the theoretical E-optimal designs and for those that show a small discrepancy, the difference quickly vanishes when we increase the flock size or the number of iterations.

It is instructive to demonstrate the search process of the PSO method in a bit more detail for this example; similar demonstrations can be shown for the other examples as well. As an illustrative example, consider the case when $a = 100$ and $b = 150$ with 128 particles and 100 iterations. Figure 1 plots the "best" maximum eigenvalue of $I(\xi, \theta)$ over the first 10 iterations of PSO procedure. Notice how quickly in just 3 iterations, PSO finds the smallest of the larger of the two eigenvalues from information

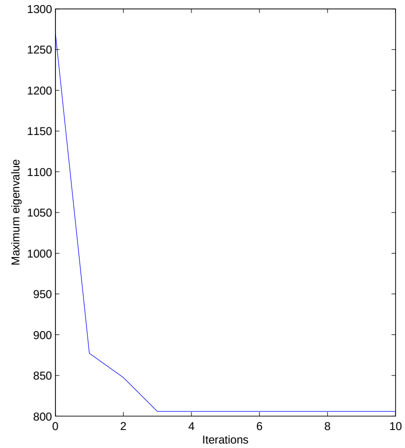


Fig. 1 Plot of the maximum eigenvalue of $I(\xi, \theta)^{-1}$ versus the number of PSO iterations in Example 1.

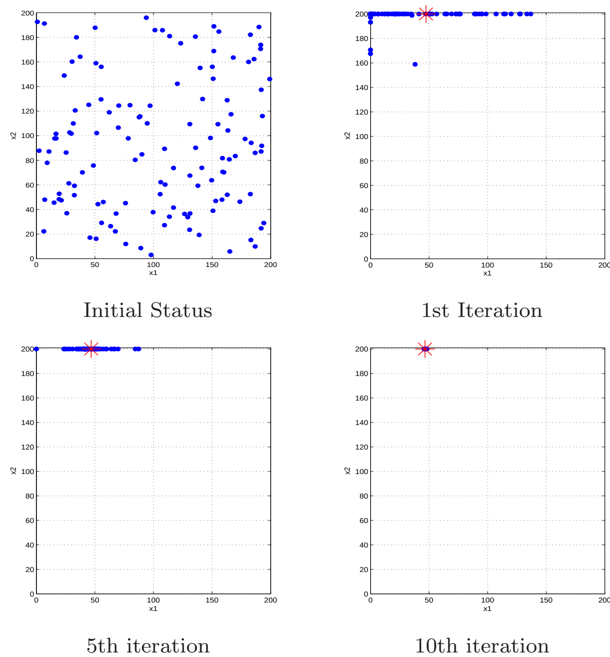
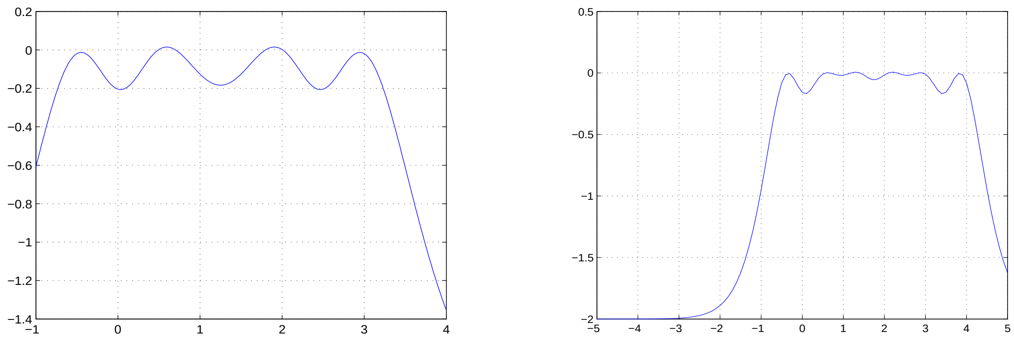


Fig. 2 The movement of particles in the PSO search for the locally E-optimal design for the Michaelis-Menten model at various stages. The red star in each of the last three plots indicates the current best design.

matrices generated by the θ's in Θ. Figure 2 shows the initial positions of the 128 randomly generated particles and how they move after the $1^{st}, 5^{th}$ and at the 10^{th} iteration when they converged.

3.2 Example 2: A minimax D-optimal design for the two-parameter logistic regression model when we have plausible ranges for the two parameters.

The widely used two-parameter logistic model assumes the probability of response is $p(x, \theta) = 1/\{1 + \exp(-b(x-a))\}$ with $\theta^\top = (a, b)$. For a given design ξ, a direct calculation shows the Fisher information



(a) (b)

Fig. 3 Plot of $c(x, \xi, \mu^*)$ versus x for Example 2 for case (a): $\Theta = [0, 2.5] \times [1, 3]$ and $X = [-1, 4]$; case (b): $\Theta = [0, 3.5] \times [1, 3.5]$ and $X = [-5, 5]$.

matrix is

$$I(\xi,\theta) = \int \begin{pmatrix} b^2 p(x,\theta)(1-p(x,\theta)) & -b(x-a)p(x,\theta)(1-p(x,\theta)) \\ -b(x-a)p(x,\theta)(1-p(x,\theta)) & (x-a)^2 p(x,\theta)(1-p(x,\theta)) \end{pmatrix} d\xi(x).$$

Suppose now that instead of having nominal values for θ, we have a priori a known set Θ of plausible values for the two parameters a and b, i.e. $\theta \in \Theta$ and Θ is known. We wish to find a minimax D-optimal design ξ^* such that

$$\xi^* = \arg\min_{\xi} \max_{\theta \in \Theta} \log(|I^{-1}(\xi,\theta)|),$$

where the minimization is over all designs on a given compact design set X. Clearly this optimal minimax design reduces to a locally D-optimal design when Θ is a singleton set.

FollowingKing and Wong (2000), we assume that $\Theta = [a_L, a_U] \times [b_L, b_U]$, where a_L, a_U, b_L and b_U are the known limits of the lower and upper bounds for a and b. In King and Wong (2000), the numerically minimax D-optimal designs were found by first running the Fedorov-Wynn algorithm (Fedorov, 1972). Invariably, the algorithm did not converge but provided clues on the number and locations of the support points of the optimal design. We then used the information along with the equivalence theorem to find the numerically optimal minimax design using Mathematica. A certain amount of guesswork was still necessary because we did not have a good understanding of the subgradient μ^*. The process was labor intensive and time consuming to find the optimal minimax design. We now use the nested PSO to find optimal minimax designs for two exemplary cases from King and Wong (2000) and compare results. For case (a), the design interval was non-symmetric and the number of particles for the inner loop is 64 and the number for the outer loop is 32. The outer iteration number was 100 and the inner iteration number was 50. In case (2), the design interval was symmetric and larger, and the number of inner particles is 256 and the number for the outer particles is 512. The outer iteration number is 200 and the inner iteration is 100. In both cases, the PSO generated designs were found quickly and a direct calculation shows both had at least 99.4% efficiency.

Case a: $\Theta = [0, 2.5] \times [1, 3]$ and $X = [-1, 4]$. The 4-point from the PSO-generated design ξ is supported at $-0.4230, 0.6164, 1.8836$ and 2.9230 and the weights at these points are $0.2481, 0.2519, 0.2519$ and 0.2481 respectively. This design is close to the one reported in King and Wong (2000) and Figure 3(a) plots $c(x, \xi, \mu^*)$ versus $x \in X$ and visually confirms the design ξ found by PSO is nearly optimal or optimal.

Case b: $\Theta = [0, 3.5] \times [1, 3.5]$ and $X = [-5, 5]$ (Example 3.2 in King and Wong (2000)). The PSO-generated 6-point design was supported at $-0.3504, 0.6075, 1.4146, 2.0854, 2.8925$ and 3.8504 and the weights at these points were $0.1799, 0.2151, 0.1050, 0.1050, 0.2151$ and 0.1799 respectively. This design is also close to the one reported in King and Wong (2000) and Figure 3(b) similarly confirms visually that the design found by PSO is nearly optimal or optimal.

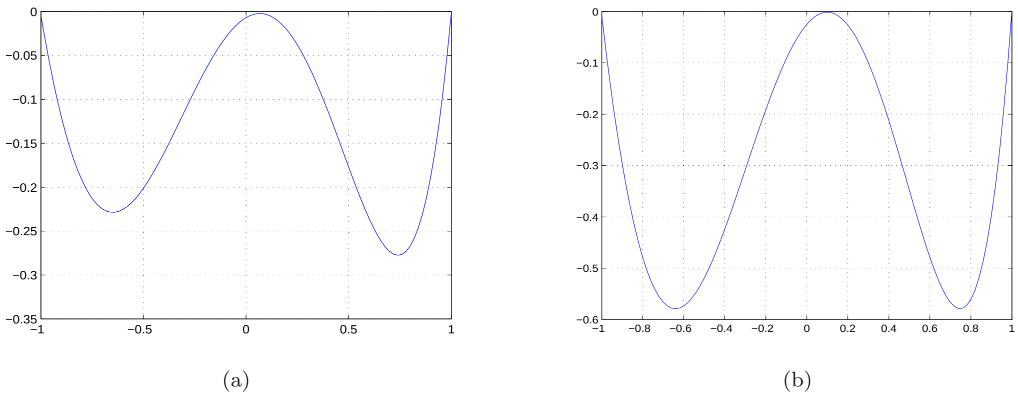


Fig. 4 Plot of $c(x, \xi, \mu^*)$ versus x over the design interval $X = [-1, 1]$ for the quadratic regression model with $\lambda(x) = 2x+5$ in Example 3 for case (a): $Z = [-1, 1]$ and case (b): $Z = [1, 1.2]$.

3.3 Example 3: A heteroscedastic minimax design for a quadratic polynomial model with an increasing efficiency function.

Consider heteroscedastic polynomial models on a given compact design space X and have the form

$$y(x) = g^\top(x)\beta + e(x)/\sqrt{\lambda(x)},$$

where $g^\top(x) = (1, x, \ldots, x^d)$, $\beta^\top = (\beta_0, \beta_1, \ldots, \beta_d)$ and $e(x)$ is a random error having mean 0 and constant variance σ^2. The function $\lambda(x)$ is a known positive real-valued continuous function defined on X and inversely proportional to the variance of the fitted response at x. All observations are assumed to be independent. As was discussed near equation (1), the variance of the fitted response at x using design ξ is proportional to $v(x, \xi) = g^\top(x)I^{-1}(\xi)g(x)$ and the sought design is ξ^* that satisfies

$$\xi^* = \arg\min_{\xi} \max_{x \in Z} v(x, \xi),$$

where the minimization is over all designs on X. Here Z is a compact set and pre-selected for prediction purposes, which may overlap with the design space X. When $Z = X$, this minimax design is called the G-optimal design (Wong and Cook, 1993). King and Wong (1998), Brown and Wong (2000) and Chen et al. (2008) proposed algorithms and discussed computational issues for finding such designs in simple and quadratic models. Our experience with these proposed algorithms is that they may not work well for more complex models and a more complicated heteroscedastic structure. Accordingly, we applied Nested PSO and tested if it can find the optimal minimax design for the quadratic model with a monotonic increasing efficiency function when (a) $X = Z$ and (b) Z is outside of X. The first case corresponds to G-optimality and the second case corresponds to a design extrapolation problem where we want to make prediction outside the design space. The optimality of the PSO-generated designs will be ascertained by equivalence theorems. In both cases, we used 128 particles and 100 iterations to find the optimal minimax designs.

Here we consider the quadratic model with a monotonic increasing efficiency function $\lambda(x) = 2x+5$. This is a more difficult problem than the case when we have a symmetric efficiency function because one can then exploit the symmetry of the design problem and reduce the dimension of the optimization problem. Specifically, we applied PSO to find an optimal minimax design when (a) $X = Z = [-1, 1]$ and (b) $X = [-1, 1]$ and $Z = [1, 1.2]$. For the first case, the PSO-generated 3-point design is supported at ± 1 and 0.0777 with weight at 1 equal to 0.2126 and weight at -1 equal to 0.4928. In the second case, the PSO-generated 3-point design is supported at ± 1 and 0.0967 with weight at 1 equal to 0.6667 and weight at -1 equal to 0.0768. The efficiency lower bounds for the PSO-generated designs are 0.9974 and 0.9975, respectively. Figure 4(a) is the graph of $c(x, \xi, \mu^*)$ for case (a) and Figure 4(b) is the graph of $c(x, \xi, \mu^*)$ for case (b). They both visually confirm the optimality of the PSO-generated designs ξ.

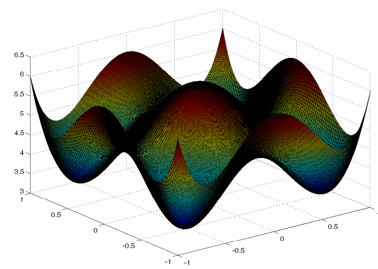


Fig. 5 Plot of the directional derivative of the 9-point PSO-generated design for the quadratic model with 2 variables.

We note that earlier work on optimal extrapolation designs for polynomial models were carried out in a series of papers by Kiefer and Wolfowitz (1964a,b, 1965); Levine (1966) assuming the efficiency function $\lambda(x)$ was a constant. Under the homoscedastic model, they were able to obtain analytic results when $X = [-1, 1]$ and $Z = [e, f]$ for selected values of e and f, including results for non-polynomial regression problems involving Chebyshev systems. Spruill (1984, 1990) worked on similar problems where bias was factored into the criterion as well. Interest in such design problems continues to date, see Broniatowski and Celant (2007) for example. PSO was able to produce optimal designs reported in the above papers and for problems with a more general setup. Because of space consideration, we do not report here additional results from PSO for extrapolation optimal minimax designs.

3.4 Example 4: A linear model with two factors

Our final example shows PSO can also find optimal minimax designs for regression models with multiple variables. Consider a homoscedastic quadratic model with two variables given by

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2$$

on the design space $(x_1, x_2) \in\ X = [-1, 1]^2$ and we want to know how to take independent observations to minimize the maximium variance of the fitted response across X. We used 500 outer iterations with an outer flock size of 500 and 50 inner iterations with an inner flock size of 50 in our PSO search. The PSO-generated design from Algorithm 2 is supported at $(-0.0296, -1)$, $(-0.0015, 0.0099)$, $(0.0161, 1)$, $(1, -0.0342)$, $(1, -1)$, $(-1, 1)$, $(1, 1)$, $(-1, -1)$ and $(-1, -0.0225)$, and the corresponding weight distribution at these points is respectively given by 0.0815,0.0962,0.0790, 0.0801,0.1454,0.1468,0.1464,0.1443 and 0.0804.

How close is this 9-point design to the G-optimal design? Figure 5 shows the plot of the directional derivative of this 9-point design and confirms visually that the the PSO-generated design is optimal or near to the G-optimal design. The model has homoscedastic errors and so the sought the G-optimal design is also the D-optimal design (Kiefer and Wolfowitz, 1960), which is always easier to find. This D or G-optimal design was reported in Farrell et al. (1968) and has weight 0.1458 at each of the 4 points $(\pm 1, \pm 1)$, weight 0.0802 at each of the 4 points $(\pm 1, 0)$ and $(0, \pm 1)$ and weight 0.0962 at the center point $(0, 0)$. The maximal variances from the D-optimal design and the PSO-generated design are 6.000 and 6.002, respectively, providing a G-efficiency of 0.9997 or 99.97% for the PSO-generated design.

In the next section, we provide computational details for the PSO. As may have been already noticed in the above examples, a couple of the designs found by the PSO method appeared to be slightly numerically different from the theoretical optimal designs. Our experience is that the discrepancy can be entirely attributed to the choices for the tuning parameters. For simplicity, we used the same set of tuning parameters for all cases in the same example even though this may not be adequate for all the cases. Generally, when more particles and more iterations are used, the discrepancy disappears and PSO is more likely able to find the optimal design. Interestingly, when we used 256 particles and 500 iterations in Example 1, the discrepancy persisted even when we increased the iteration and particle numbers to the thousands. Further investigation revealed that the smaller support point of the theoretical optimal design in the first row of Table 1 calculated from the formula was wrongly reported and the correct value was the one found by PSO!

4 Computational and Implementation Details for PSO

Particle swarm optimization (PSO), proposed by Eberhart and Kennedy (1995), is an iterative method that can be generically and readily coded to simulate the behavior of a flock of bird in search for food. Before presenting our modified PSO algorithm for finding optimal minimax designs, we first describe how PSO works in its most basic for solving a minimization problem:

$$\min_{\mathbf{x}\in X} f(\mathbf{x}), \tag{2}$$

where X is a given compact domain and $f(\mathbf{x})$ is the objective function. We initialize PSO using a user-specified number, say n, randomly generated particles to search for the optimum over the search space. In our context, X is the design space, $f(\mathbf{x})$ is the design criterion formulated as a convex function of the information matrix and the particles are the flock of birds or search designs defined by their mass distributions and support points. If the model has k parameters in the mean function, it is typical to choose the initial flock as search designs all with k support points.

The two basic equations that drive movement for the i^{th} particle in the PSO algorithm in its search to optimize an objective function $f(\mathbf{x})$ is as follows. At times t ant $t+1$, the movement of particle i is governed by

$$\mathbf{w}_i^{t+1} = \theta_t \mathbf{w}_i^t + \gamma_1 \alpha_1 (\mathbf{p}_i - \mathbf{x}_i^t) + \gamma_2 \alpha_2 (\mathbf{p}_g - \mathbf{x}_i^t), \tag{3}$$

and

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{w}_i^{t+1}. \tag{4}$$

Here, $\mathbf{w}_i^t$ and $\mathbf{x}_i^t$ are, respectively, the velocity and the current position for the i^{th} particle at time t. The inertia weight θ_t modulates the influence of the former velocity and can be a constant or a decreasing function with values between 0 and 1. For example, Eberhart and Shi (2000) used a linearly decreasing function over the specified time range with an initial value 0.9 and end value of 0.4. The vector $\mathbf{p}_i$ is the personal best (optimal) position attained by the ith particle up to time t and the vector $\mathbf{p}_g$ is the global best (optimal) position attained among all particles up to time t. This means that up to time t, the personal best for particle i is $pbest_i = f(\mathbf{p}_i)$ and $gbest = f(\mathbf{p}_g)$. The two random vectors in the PSO algorithm are α_1 and α_2 and their components are usually taken to be independent random variables from $U(0,1)$. The constant γ_1 is the cognitive learning factor and γ_2 is the social learning factor. These two constants determine how each particle moves toward its own personal best position or overall global best position. The default values for these two constants in the PSO codes are $\gamma_1 = \gamma_2 = 2$ and they really seem to work well in practice for nearly all problems that we have investigated so far. Note that in equation (3), the product in the middle two terms is Hadamard product. Further details are in Chatterjee and Siarry (2006); Fan and Chang (2007); Shi and Eberhart (1998a) and Shi and Eberhart (1998b).

The particles' movement along various paths are clamped to a user-specified maximum velocity $\mathbf{w}_{max}$. After updating the velocity $\mathbf{w}_i$ via (3), if a certain component of $\mathbf{w}_i$ exceeds the corresponding component of $\mathbf{w}_{max}$, the component velocity will be limited to the corresponding component value of $\mathbf{w}_{max}$. In our implementation, we set $\mathbf{w}_{max} = 100 \cdot \mathbf{1}$, where $\mathbf{1}$ is the unit vector.

Algorithm 1 ***PSO for the minimization problem*** (2)

(A1a)	Initialize *particles*	
	(A1a.1)	Choose *initial position* $\mathbf{x}_i$ *and velocity* $\mathbf{w}_i$ *for particle i, for* $i = 1, \ldots, n$.
	(A1a.2)	Calculate *fitness values* $f(\mathbf{x}_i)$.
	(A1a.3)	Determine *local and global best positions* $\mathbf{p}_i$ *and* $\mathbf{p}_g$.
(A1b)	Repeat until *stopping criteria are satisfied.*	
	(A1b.1)	Calculate *each particle velocity using equation* (3).
	(A1b.2)	Update *each particle position using equation* (4).
	(A1b.3)	Calculate *fitness values* $f(\mathbf{x}_i)$.
	(A1b.4)	Update *best positions* $\mathbf{p}_i$ *and* $\mathbf{p}_g$ *and best values* $pbest_i$ *and gbest.*
(A1c)	Output $\mathbf{p}_g$ *and gbest.*	

To find optimal minimax designs, we modify Algorithm 1 to enable multiple particles to calculate the fitness $f_{outer}(v_i)$ values simultaneously. We call the modified PSO method Nested Particle Swarm Optimization because it involves double optimization, one after the other. More generally, let $g(u, v)$ be a given function defined on two compact spaces $\mathcal{U}$ and $\mathcal{V}$. Minimax optimization problems have the form:

$$\min_{\mathbf{u}\in\mathcal{U}} \max_{\mathbf{v}\in\mathcal{V}} g(\mathbf{u}, \mathbf{v}) \equiv \min_{\mathbf{u}\in\mathcal{U}} f_{outer}(\mathbf{u}) \equiv \min_{\mathbf{u}\in\mathcal{U}} \left[-\min_{\mathbf{v}\in\mathcal{V}} f_{inner}(\mathbf{v}) \right], \tag{5}$$

where

$$f_{outer}(\mathbf{u}) = \max_{\mathbf{v}\in\mathcal{V}} g(\mathbf{u}, \mathbf{v}), \tag{6}$$

and for fixed $\mathbf{u}$,

$$f_{inner}(\mathbf{v}) = -g(\mathbf{u}, \mathbf{v}). \tag{7}$$

We call functions $f_{outer}(\mathbf{u})$ and $f_{inner}(\mathbf{v})$ the outer and inner objective functions respectively. For our design problems, we set $g(u, v), \mathcal{U}$ and $\mathcal{V}$ appropriately. For instance, for G-optimality, we let $\mathcal{U}$ be the set of all designs defined on X, let $\mathcal{V} = X$ and let $g(\mathbf{u}, \mathbf{v})$ be the variance of the fitted response at $\mathbf{v}$ for design $\mathbf{u}$. The same setup is used for Example 3, except that we now replace $\mathcal{V} = X$ by $\mathcal{V} = Z$. The minimax design problem is now formulated as a nested (or double) minimization problem and solved using Algorithm 2, which in essence is Algorithm 1 applied twice, once to the outer function and another to the inner function.

Algorithm 2 ***Nested PSO for the minimax problem*** (5).

(A2a)	Initialize *particles*	
	(A2a.1)	Choose *initial position* $\mathbf{x}_i$ *and velocity* $\mathbf{w}_i$ *for particle i, for* $i = 1, \ldots, n$.
	(A2a.2)	Calculate *fitness values* $f_{outer}(\mathbf{x}_i)$ *by solving* (7) *by Algorithm 1.*
	(A2a.3)	Determine *local and global best positions* $\mathbf{p}_i$ *and* $\mathbf{p}_g$.
(A2b)	Repeat until *stopping criteria are satisfied.*	
	(A2b.1)	Calculate *each particle velocity using equation* (3).
	(A2b.2)	Update *each particle position using equation* (4).
	(A2b.3)	Calculate *fitness values* $f_{outer}(\mathbf{x}_i)$ *by solving* (7) *by Algorithm 1.*
	(A2b.4)	Update *best positions* $\mathbf{p}_i$ *and* $\mathbf{p}_g$ *and best values* $pbest_i$ *and gbest.*
(A2c)	Output $\mathbf{p}_g$ *and gbest.*	

To apply the Nested PSO to solve minimax design problems we use Example 3 as an illustrative example and set $f_{outer}(\xi) = \max_{z\in Z} v(z, \xi)$ which is first computed via PSO for each fixed ξ. The optimal design is then found by another PSO by treating each particle as a design ξ represented as $(x_1, \ldots, x_k, p_1, \ldots, p_k)^\top$, where $x_i, i = 1, \ldots, k$ are the support points in the design space and p_i, $i = 1, \ldots, k$ are the corresponding weights with $1 > p_i > 0$ and $\sum_{i=1}^k p_i = 1$.

All optimal minimax designs in Section 3 were found using Algorithm 2. In the supplementary material, we provide open PSO codes that implement Algorithm 2 and demonstrate how to use a MATLAB toolbox to obtain a G-optimal design for an illustrative case when we have a simple linear model and the efficiency function is $\lambda(x) = x + 5$ defined on $X = [-1, 1]$. We also show how the codes can be readily amended to find different optimal designs under various setups.

5 Standardized Maximin Optimal Designs for Enzyme Inhibition Kinetic Models

The two-parameter Michaelis-Menten model in Example 1 is commonly used enzyme kinetics studies. There are four popular extensions of the Michaelis-Menten model used to further identify the types of inhibition process involved in the enzyme-inhibitor system. These nonlinear models have three or four parameters and their meann functions are

Competitive Inhibition Model:

$$\upsilon = \frac{VS}{K_m(1 + \frac{I}{K_{ic}}) + S}; \tag{8}$$

Noncompetitive Inhibition Model:

$$\upsilon = \frac{VS}{(K_m + S)(1 + \frac{I}{K_{ic}})}; \tag{9}$$

Uncompetitive Inhibition Model:

$$\upsilon = \frac{VS}{K_m + S(1 + \frac{I}{K_{iu}})}; \tag{10}$$

Mixed Inhibition Model:

$$\upsilon = \frac{VS}{K_m(1 + \frac{I}{K_{ic}}) + S(1 + \frac{I}{K_{iu}})}. \tag{11}$$

Here S and I are the two design variables denoting the concentration of the substrate and the inhibitor concentration respectively. The model parameters are V, K_m, K_{ic}, K_{iu}, and Bogacka et al. (2011) found locally D-optimal designs for these four enzyme inhibition kinetic models. The locally D-optimal designs do not depend on V because this parameter enters the four models linearly. Thus we only consider the parameter vector $\theta = (K_m, K_{ic}, K_{iu})^\top$.

We now use the nested PSO algorithm to find standardized maximin D-optimal designs for these models. Such a design criterion was used in Dette and Biedermann (2003) to robustify locally optimal designs against mis-specification of the nominal values in the Michaelis-Menten model. Specifically, let ξ_θ^* be the locally D-optimal design with respect to the parameter θ and let Θ be a known set containing plausible values of θ. The goal here is to seek an optimal design that maximizes the design criterion $\Phi(\xi)$, where

$$\Phi(\xi) = \min_{\theta \in \Theta} \frac{|I(\xi, \theta)|}{|I(\xi_\theta^*, \theta)|}.$$

We follow the set up in Bogacka et al. (2011) where the design space for the two variables $\mathbf{x} = (S, I)$ is $X = [0, 30] \times [0, 60]$ and the range set Θ of possible values for $\theta = (K_m, K_{ic}, K_{iu})^\top$ is $[4, 5] \times [2, 3] \times [4, 5]$, which includes the nominal values used in their study for an application using the Competitive Inhibition model. The nested PSO-generated standardized maximin optimal designs are shown in Table 2 along with their efficiency lower bounds. For each of these design ξ, the bound is given by

$$\frac{p}{\max_X \int_{A(\xi)} \frac{\partial g(x,\theta)^\top}{\partial \theta} I(\xi, \theta)^{-1} \frac{\partial g(x,\theta)}{\partial \theta} \mu(d\theta)},$$

where p is the number of the parameters in the mean function, $A(\xi) = \{\theta \in \Theta | \Phi(\xi) = \text{eff}_\theta(\xi)^2\}$, $\text{eff}_\theta(\xi) = (|I(\xi, \theta)|/|I(\xi_\theta^*, \theta)|)^{1/2}$ and μ is the probability measure defined on $A(\xi)$ that minimizes the denominator; see Wong and Cook (1993) or Dette and Biedermann (2003) for details. For example, we have $p = 4$ for the mixed inhibition model. Table 2 shows that all the designs found by the nested PSO are at least 99.9% efficient and so they are all very close to the theoretical standardized maximin optimal designs.

To find the maximin optimal designs, one notes that the maximin criterion is a concave function on the space of designs on X and so conditions from an equivalence theorem can be applied. For example,

Table 2 The nested PSO-generated standardized maximin D-optimal designs for the four inhibition models using the following PSO parameters: number of particles in the outer(inner) loop=256(128), number of iterations in the outer(inner) loop=200(100) and $\gamma_1 = \gamma_2 = 2$)

Type	ξ_{PSO}				Lower bound of efficiency	Maximum point of dispersion function
Competitive Inhibition Model	$\begin{pmatrix} 3.4445 \\ 0.0000 \end{pmatrix}$ 0.3333	$\begin{pmatrix} 30.0000 \\ 0.0000 \end{pmatrix}$ 0.3333	$\begin{pmatrix} 30.0000 \\ 18.8982 \end{pmatrix}$ 0.3334		99.99%	3.0003
Noncompetitive Inhibition Model	$\begin{pmatrix} 3.4429 \\ 0.0000 \end{pmatrix}$ 0.3333	$\begin{pmatrix} 30.0000 \\ 0.0000 \end{pmatrix}$ 0.3333	$\begin{pmatrix} 30.0000 \\ 2.4495 \end{pmatrix}$ 0.3334		99.99%	3.0003
Uncompetitive Inhibition Model	$\begin{pmatrix} 3.4461 \\ 0.0000 \end{pmatrix}$ 0.3333	$\begin{pmatrix} 30.0000 \\ 0.0000 \end{pmatrix}$ 0.3334	$\begin{pmatrix} 30.0000 \\ 5.1383 \end{pmatrix}$ 0.3333		99.99%	3.0003
Mixed Inhibition Model	$\begin{pmatrix} 3.4406 \\ 0.0000 \end{pmatrix}$ 0.2503	$\begin{pmatrix} 4.2835 \\ 3.1445 \end{pmatrix}$ 0.2498	$\begin{pmatrix} 30.0000 \\ 0.0000 \end{pmatrix}$ 0.2501	$\begin{pmatrix} 30.0000 \\ 4.0191 \end{pmatrix}$ 0.2498	99.92%	4.0033

Table 3 Standardized maximin D-optimal designs for the four kinds of inhibition models.

Type	ξ^*			
Competitive Inhibition Model	$\begin{pmatrix} 3.4429 \\ 0 \end{pmatrix}$ 1/3	$\begin{pmatrix} 30 \\ 0 \end{pmatrix}$ 1/3	$\begin{pmatrix} 30 \\ 18.8944 \end{pmatrix}$ 1/3	
Noncompetitive Inhibition Model	$\begin{pmatrix} 3.4429 \\ 0 \end{pmatrix}$ 1/3	$\begin{pmatrix} 30 \\ 0 \end{pmatrix}$ 1/3	$\begin{pmatrix} 30 \\ 2.4495 \end{pmatrix}$ 1/3	
Uncompetitive Inhibition Model	$\begin{pmatrix} 3.4429 \\ 0 \end{pmatrix}$ 1/3	$\begin{pmatrix} 30 \\ 0 \end{pmatrix}$ 1/3	$\begin{pmatrix} 30 \\ 5.1424 \end{pmatrix}$ 1/3	
Mixed Inhibition Model	$\begin{pmatrix} 3.4429 \\ 0 \end{pmatrix}$ 1/4	$\begin{pmatrix} 4.2943 \\ 3.1231 \end{pmatrix}$ 1/4	$\begin{pmatrix} 30 \\ 0 \end{pmatrix}$ 1/4	$\begin{pmatrix} 30 \\ 4.0199 \end{pmatrix}$ 1/4

consider the competitive inhibition model where Table 2 shows the optimal design ξ^*_{ci} should be an equally weighted design with the following structure:

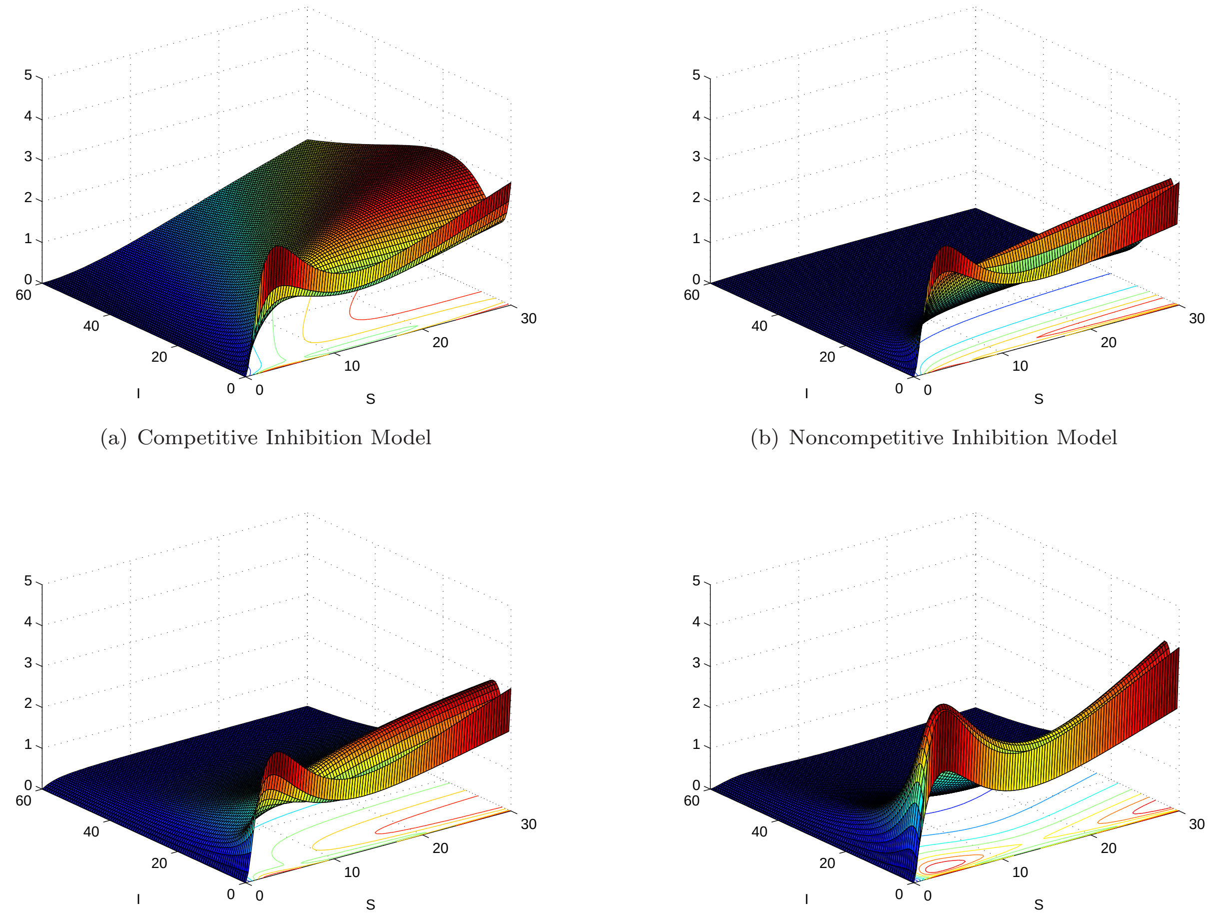
$$\xi^*_{ci} = \begin{pmatrix} \begin{pmatrix} S_1 \\ 0 \end{pmatrix} & \begin{pmatrix} 30 \\ 0 \end{pmatrix} & \begin{pmatrix} 30 \\ I_3 \end{pmatrix} \\ 1/3 & 1/3 & 1/3 \end{pmatrix}. \tag{12}$$

One could conjecture that when we have the maximin optimal design ξ^*, the measure μ^* is equally supported at $(4,3)^\top$ and $(5,2)^\top$ in the parameter space and the requirements of the equivalence theorem provide us with equations to solve for S_1 and I_3. In this case, the solutions are $S_1 = 3.4429$ and $I_3 = 18.8944$. Both values are close to the design points shown in Table 2 and the design displayed in Table 3 for this model is the standardized maximin optimal design. Similarly, the other designs found using hints from the generated designs shown in Table 3 are also standardized maximin optimal for the other 3 models. The plots of the directional derivatives for the maximin criterion for these 4 designs in Figure 6 confirm their optimality.

6 Discussion

In today rapidly rising cost of experimentation, optimal design ideas take on an increasingly important role. A well designed study is able to answer the scientific questions accurately and with minimum cost. It is therefore not surprising that optimal experimental designs continue to find increasingly more applications in different fields and novel applications are continually seen in traditional areas, see Berger and Wong (2005), for example.

Computer algorithms have played and will continue to play an important role in our search of optimal designs. They are usually sequential in nature and typically involve the addition of a carefully selected new design point to the current design by mixing them appropriately to form a new design.



(a) Competitive Inhibition Model

(b) Noncompetitive Inhibition Model

(c) Uncompetitive Inhibition Model

(d) Mixed Inhibition Model

Fig. 6 The directional derivatives of the standardized maximin D-optimal designs for four inhibition models.

The generated design accumulates many points or clusters of points as the algorithm proceeds and judicious rules for collapsing points into distinct points is required. The sequence of weights used to form a new design at each iteration is chosen so that they are all between 0 and 1, have a finite sum but does not converge too quickly or prematurely. Stopping rules are employed to decide when to terminate the search; they typically require either a maximum number of iterations allowed or when the change in the value of the optimality criterion in successive searches is negligible according to a user-selected tolerance level. An example of such an algorithm is the noted Fedorov-Wynn algorithm which is still popular after more than 3 decades of use. Details and exemplary codes for generating D- and c-optimal designs can be found in design monographs like Silvey (1980) and Fedorov (1972). Several modified versions of the Fedorov-Wynn algorithms have been proposed and we refer to them as the Fedorov-Wynn types of algorithms.

A main difference between PSO and the popular Fedorov-Wynn types of algorithm is that PSO uses many particles (designs) right from the start to cover the design space before searching for the optimum, whereas the Fedorov-Wynn types of algorithms use only one starting design. This means that a poor choice of the starting design in the Fedorov-Wynn algorithm may require a relatively long time for it to get near the optimum. In contrast, PSO uses many particles to search for the optimum at any one time by sharing information among the search particles. In addition, PSO is flexible and easy to implement; our experience is that only the number of iterations and flock size seem to affect

PSO ability to find the optimal design.; all other tuning parameters in the PSO do not seem to matter, and so we set them all equal to their default values. In this sense, PSO compares favorably with other algorithms like the genetic algorithm that can depend sensitively on all the tuning parameters. In sum, our experience with PSO agrees with findings reported in the literature.

To get a sense of computing time that Nested PSO required to run through a search, we revisit Example 1 for the Michaelis-Menten model. For brevity, we consider an illustrative case when the model parameters are $(a, b)^\top = (100, 150)$ and we use different numbers of particles and iterations. When the iteration number is fixed at 100, and the number of particles is 128, 256, 512, 1024 and 2048, the search took 0.87, 1.65, 3.16, 6.32, 12.58 seconds respectively. When the number of particles is fixed at 128, and the iteration number is 200, 500 and 1000, the PSO search time is 1.68, 4.13 and 8.05 seconds respectively. In all cases, the generated designs agree up to 5 decimal places in terms of both weights and design points. Clearly larger flock size requires more time to partake in the sharing of information and larger number of iterations requires longer time.

In summary, PSO is a novel and powerful method to generate different types of optimal experimental designs. We continue to have other successes not reported here when we applied PSO to find A, c or D-optimal designs for nonlinear models with 3 or more parameters. Each time PSO would find and confirm the results in the literature usually in a few seconds of CPU time. We have also verified that PSO is able to generate the D-optimal designs for Scheffe's quadratic polynomial mixture models up to 8 factors where there are more than a hundred variables to be optimized.

PSO methodology has potential for finding other types of optimal designs. We have two areas for future work. The first is to apply PSO to find multiple-objective optimal designs. Such designs are more attractive because studies typically have several goals and not of them may be of equal interest. Multiple-objective optimal designs are discussed extensively with examples in Cook and Wong (1994); Huang and Wong (1998); Zhu and Wong (2000) and Zhu and Wong (2001). The second area for future work is to apply PSO to find optimal designs under a non-convex criterion, where we no longer have an equivalence theorem to confirm the design optimality. Our latest results include modifying PSO in a novel way to find balanced optimal supersaturated designs which have a very different set up than the one considered here. The optimization problem is high dimensional as we consider having more factors than has been considered in the literature (Phoa el at., 2013). other examples of optimal designs under a non-convex objective functions are exact optimal design, replication free optimal designs, minimum bias designs or designs that minimize the mean square error. We plan to apply PSO methodology to find these types of optimal designs and hope to report results in the near future.

Acknowledgements Weng Kee Wong worked on the manuscript when he was a visiting fellow at The Sir Isaac Newton Institute at Cambridge, England and a scientific advisor for a six-month workshop in the design and analysis of experiment. He would like to thank Professor Rosemary Bailey for hosting the workshop and the Institute for the support during his repeated visits there in the second half of 2011. The research of Ray-Bing Chen was partially supported by the National Science Council of Taiwan and the Mathematics Division of the National Center for Theoretical Sciences (South) in Taiwan. The research of Weichung Wang was partially supported by the National Science Council of Taiwan and the Taida Institute of Mathematical Sciences.

References

Atkinson, A. C., Donev, A. N., and Tobias, R. D. (2007). *Optimum Experimental Designs, with SAS.* Oxford University Press.

Berger, M. P. F., King, J., and Wong, W. K. (2000). Minimax D-optimal Designs for Item Response Theory Models. *Psychometrika*, 65(3):377–390.

Berger, M. P. F. and Wong, W. K. (2005). *Applied Optimal Designs.* Wiley.

Bogacka, B., Patan, M., Johnson, P. J., Youdim, K., and Atkinson, A. C. (2011). Optimum Design of Experiments for Enzyme Inhibition Kinetic Models. *Journal of Biopharmaceutical Statistics*, 21(3):555–572.

Broniatowski, M. and Celant, G. (2007). Optimality and Bias of Some Interpolation and Extrapolation Designs. *Journal of Statistical Planning and Inference*, 137(3):858–868.

Brown, L. D. and Wong, W. K. (2000). An Algorithmic Construction of Optimal Minimax Designs for Heteroscedastic Linear Models. *Journal of Statistical Planning and Inference*, 85(1-2):103–114.

Chatterjee, A. and Siarry, P. (2006). Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization. *Computers and Operations Research*, 33(3):859–871.

Chen, R. B., Wong, W. K., and Li, K. Y. (2008). Optimal Minimax Designs over a Prespecified Interval in a Heteroscedastic Polynomial Model. *Statistics and Probability Letters*, 78(13):1914–1921.

Clerc, M. (2006). *Particle Swarm Optimization*. Wiley-ISTE.

Cook, R. D. and Wong, W. K. (1994). On the Equivalence of Constrained and Compound Optimal Designs. *Journal of the Americian Statistical Association*, 89:687–692.

Dette, H. and Biedermann, S. (2003). Robust and Efficient Designs for the Michaelis-Menten Model. *Journal of the American Statistical Association*, 98(463):679–686.

Dette, H. and Wong, W. K. (1999). E-optimal Designs for the Michaelis-Menten Model. *Statistics and Probability Letters*, 44(4):405–408.

Eberhart, R. and Kennedy, J. (1995). A New Optimizer Using Particle Swarm Theory. In *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*, pages 39–43. IEEE.

Eberhart, R. C. and Shi, Y. (2000). Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 84–88. IEEE.

Fan, S. K. S. and Chang, J. M. (2007). A Modified Particle Swarm Optimizer Using an Adaptive Dynamic Weight Scheme. In *Proceedings of the 1st International Conference on Digital Human Modeling*, pages 56–65. Springer-Verlag.

Farrell, E. H., Kiefer, J., and Walbran, A. (1968). Optimum Multivariate Designs. In *Proceedings of the Fifth Berkeley Symposium on Probability and Mathematical Statistics, Volume 1, L. M. Le Cam and J. Neyman (eds)*, pages 113–138. University of California Press.

Fedorov, V. V. (1972). *Theory of Optimal Experiments*. Academic press.

Huang, Y. C. and Wong, W. K. (1998). Multiple-objective Optimal Designs. *Journal of Biopharmaceutical Statistics*, 8:635–643.

Johnson, R. T., Montgomery, D. C., and Jones, B. A. (2011). An Expository Paper on Optimal Design. *Quality Engineering*, 23(3):287–301.

Kiefer, J. (1985). *Collected Papers: Design of Experiments*. Springer.

Kiefer, J. and Wolfowitz, J. (1960). The Equivalence of Two Extremum Problems. *The Canadian Journal of Mathematics*, 12(3):363–366.

Kiefer, J. and Wolfowitz, J. (1964a). Optimum Extrapolation and Interpolation Designs I. *Annals of the Institute of Statistical Mathematics*, 16(1):79–108.

Kiefer, J. and Wolfowitz, J. (1964b). Optimum Extrapolation and Interpolation Designs II. *Annals of the Institute of Statistical Mathematics*, 16(1):295–303.

Kiefer, J. and Wolfowitz, J. (1965). On a Theorem of Hoel and Levine on Extrapolation Designs. *The Annals of Mathematical Statistics*, 36(6):1627–1655.

King, J. and Wong, W. K. (1998). Optimal Minimax Designs for Prediction in Heteroscedastic Models. *Journal of Statistical Planning and Inference*, 69(2):371–383.

King, J. and Wong, W. K. (2000). Minimax D-optimal Designs for the Logistic Model. *Biometrics*, 56(4):1263–1267.

Levine, A. (1966). A Problem in Minimax Cariance Polynomial Extrapolation. *The Annals of Mathematical Statistics*, 37(4):898–903.

Noubiap, R. F. and Seidel, W. (2000). A Minimax Algorithm for Constructing Optimal Symmetrical Balanced Designs for a Logistic Regression Model. *Journal of Statistical Planning and Inference*, 91:151–168.

Pázman, A. (1986). *Foundations of Optimum Experimental Design*. Springer-Verlag.

Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle Swarm Optimization: An Overview. *Swarm Intell*, 1:33–57.

Rodriguez, M., Jones, B., Borror, C. M., and Montgomery, D. C. (2010). Generating and Assessing Exact G-optimal Designs. *J. of Quality Technology*, 42:3–20.

Shi, Y. and Eberhart, R. (1998a). A Modified Particle Swarm Optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 69–73. IEEE.

Shi, Y. and Eberhart, R. (1998b). Parameter Selection in Particle Swarm Optimization. In *Evolutionary Programming VII*, pages 591–600. Springer.

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

Sitter, R. R. (1992). Robust Designs. *Biometrics*, 48:1145–1155.

Spruill, M. C. (1984). Optimal Designs for Minimax Extrapolation. *Journal of Multivariate Analysis*, 15(1):52–62.

Spruill, M. C. (1990). Optimal Designs for Multivariate Interpolation. *Journal of Multivariate Analysis*, 34(1):141–155.

Whitacre, J. M. (2011a). Recent Trends Indicate Rapid Growth of Nature-Inspired Optimization in Academia and Industry. *Computing*, 93:121–133.

Whitacre, J. M. (2011b). Survival of the Flexible: Explaining the Recent Dominance of Nature-Inspired Optimization Within a Rapidly Evolving World. *Computing*, 93:135–146.

Wong, W. K. (1992). A Unified Approach to the Construction of Minimax Designs. *Biometrika*, 79(3):611.

Wong, W. K. and Cook, R. D. (1993). Heteroscedastic G-optimal Designs. *Journal of the Royal Statistical Society, Ser. B*, 55(4):871–880.

Yang, X. S. (2010). *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley.

Zhu, W. and Wong, W. K. (2000). Dual-objective Bayesian Optimal Designs for a Dose-Ranging Study. *Drug Information Journal*, 34:421–428.

Zhu, W. and Wong, W. K. (2001). Bayesian Optimal Designs for Estimating a Set of Symmetric Quantiles. *Statistics in Medicine*, 20:123–137.