Published in final edited form as:

Swarm Evol Comput. 2014 October 1; 18: 1–10. doi:10.1016/j.swevo.2014.06.003.

Using Animal Instincts to Design Efficient Biomedical Studies via Particle Swarm Optimization

Jiaheng Qiu^a, Ray-Bing Chen^b, Weichung Wang^c, and Weng Kee Wong^d

^aDepartment of Biostatistics, University of California, Los Angeles, CA 90095, US

^bDepartment of Statistics, National Cheng-Kung University, Tainan 70101, Taiwan

^cDepartment of Mathematics, National Taiwan University, Taipei, Taiwan

^dDepartment of Biostatistics, University of California, Los Angeles, CA 90095, USA

Abstract

Particle swarm optimization (PSO) is an increasingly popular metaheuristic algorithm for solving complex optimization problems. Its popularity is due to its repeated successes in finding an optimum or a near optimal solution for problems in many applied disciplines. The algorithm makes no assumption of the function to be optimized and for biomedical experiments like those presented here, PSO typically finds the optimal solutions in a few seconds of CPU time on a garden-variety laptop. We apply PSO to find various types of optimal designs for several problems in the biological sciences and compare PSO performance relative to the differential evolution algorithm, another popular metaheuristic algorithm in the engineering literature.

Keywords

Approximate design; *c*-optimal design; *D*-optimal design; efficiency; metaheuristic algorithms; particle swarm optimization

1. Introduction

Optimal experimental designs have been gaining attention in the last two decades [1]. A main reason is rising cost in conducting experiments and an increasing realization in more applied fields that optimal design ideas can save costs substantially without sacrifice in statistical efficiency. Some examples are given in [2], [3], [4], [5] and [6], where the problems include designing reaction kinetics studies to medical studies with a time-to-event outcome. Berger and Wong [7] describes a collection of concrete applications of optimal designs to real problems that ranges from applications in biomedical and social science arenas, including a design problem to identify optimal locations to monitor groundwater wells in the Los Angeles basin.

© 2014 Elsevier B.V. All rights reserved.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Nonlinear models are frequently used to study outcomes or responses in biomedical experiments. This means that we assume a known nonlinear functional relationship between the mean response and the independent variables. This function has unknown parameters that determine the shape and properties of the mean response and one common goal in the study is to estimate the parameters in the mean function. In addition, the model also has an unobservable error term with mean zero and constant variance. Given a study objective, the design problem involves selecting the right number of combination levels of the independent variables to observe the outcome and what these levels are. The design optimality criterion for nonlinear models depends on the values of the model parameters and nominal values (or best guesses for these parameters) are required before the optimal design can be implemented. Because the optimal designs depend on the nominal values, they are termed locally optimal. Such optimal designs usually represent the first step in finding an optimal design strategy and is the simplest to construct and study.

The analytical description of the locally optimal design for a nonlinear model is rarely available unless the model is very simple. When they do exist, they are usually complicated; see for example, the analytical description for the locally D -optimal design for estimating the two parameters in the logistic model [8]. Further, the formula or analytical description of the optimal design in a nonlinear model is invariably derived under a set of mathematical assumptions that may or may not apply in practice. For these reasons, it is desirable to have a flexible and effective algorithm that can find a variety of optimal designs quickly and reliably.

There are algorithms for finding optimal designs and most are based on heuristics or intuition and they do not have a theoretical basis. Only a couple of algorithms can be proven to converge to the optimal designs and prominent ones include Fedorov's and Wynn's algorithms for generating D and c -optimal designs [9, 10]. The former designs are useful for estimating all parameters in the mean function and the latter targets estimation of a specific function of the model parameters by minimizing, respectively, the volume of the confidence ellipsoid and the asymptotic variance of the estimated function of interest. For the few algorithms that can be shown to converge mathematically, problems may still exist and they include (i) they take too long to converge, (ii) they may fail to converge for more complicated setups that they are not designed, such as for nonlinear for mixed effects nonlinear models, and (iii) numerical issues due to rounding problems or the intrinsic nature of the sequential process; for example, many algorithms produce clusters of support points as the algorithm proceeds and these clusters require periodic and judicious collapsing into the correct distinct but unknown support points.

In the next section, we briefly review particle swarm optimization (PSO) methodology and show it is an exciting, easy and effective algorithm to generate optimal designs for statistical models. This algorithm has been used for almost a dozen of years in the computer science and engineering circles, and increasingly more so in recent years due to its repeated successes in solving an increasing large class of applied problems. The main reasons for its popularity seem to be its flexibility, ease of implementation and utility, and general applicability to solve (or nearly solve) complex optimization problems without having to make specific assumptions on the objective function. In Section 3 we present the statistical

background and demonstrate PSO can efficiently find a variety of optimal designs for different types of nonlinear models in biomedicine. Section 4 provides a discussion and also compares PSO performance relative to the differential evolution algorithm, which is another popular metaheuristic algorithm for solving optimization problems in engineering problems. Section 5 is the conclusion.

2. Particle Swarm Optimization (PSO)

Nature-inspired algorithms have been gaining popularity and dominance in the last decade both in academia and industrial applications after adjusting for different types of biases [11, 12]. One of the most prominent examples of a nature-inspired algorithm is Particle Swarm Optimization (PSO) based on swarm intelligence. It is a metaheuristic algorithm and comes about from the research in fish and swarm movement behavior. PSO is intriguing in that they always seem to be able to quickly solve the optimization problem or provide good quality solutions for many types of complex optimization problems, even though the method itself lacks a firm theoretical justification to date. An attempt to explain its success from a biological viewpoint is given in Garnier, et al. [13] and an overview of PSO is available in Poli et al. [14]. Common characteristics of PSO includes its ability to find the optimal solution to a complex problem or gets close to the optimal solution quickly without requiring any assumption on the function to be optimized. PSO codes are available on many websites such as <http://www.swarmintelligence.org>, <http://particleswarm.info> with the latter website more updated and consistently so. In addition, codes are available in books on metaheuristic methods like Yang [15] and also in software such as MATLAB where several PSO codes can be found at the Programs section at <http://www.mathworks.com/matlabcentral/fileexchange/7506>.

There are two basic equations that drive movement for the each particle in the PSO algorithm in its search to optimize an objective function $h(\cdot)$. At times t and $t + 1$, the movement of particle i is governed by

$$\mathbf{v}_i^{t+1} = \tau_t \mathbf{v}_i^t + \gamma_1 \beta_1 \odot (\mathbf{p}_i - \mathbf{x}_i^t) + \gamma_2 \beta_2 \odot (\mathbf{p}_g - \mathbf{x}_i^t), \quad (1)$$

and

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}. \quad (2)$$

Here, $\mathbf{v}_i^t$ is the particle velocity at time t and $\mathbf{x}_i^t$ is the current particle position at time t . The inertia weight τ_t adjusts the influence of the former velocity and can be a constant or a decreasing function with values between 0 and 1. For example, a linearly decreasing function over the specified time range with initial value 0.9 and end value 0.4 [16]. Further, the vector $\mathbf{p}_i$ is the personal best (optimal) position as perceived by the i th particle and the vector $\mathbf{p}_g$ is the global best (optimal) position as perceived by all particles, up to time t . This means that up to time t , the personal best for particle i is $pbest_i = h(\mathbf{p}_i)$ and $gbest = h(\mathbf{p}_i)$. The two random vectors in the PSO algorithm are β_1 and β_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. For our design problems, we set $\gamma_1 = \gamma_2 = 2$ and they all worked well for the problems we discussed including almost all other design problems that we have investigated this far. In equation (1), the product in the last two terms is Hadamard product.

The pseudo code for the PSO procedure for a flock of size n is as follows.

```

1  Initialize particles
    (1.1) Initiate positions  $x_i$  and velocities  $v_i$  for  $i = 1, \dots, n$ .
    (1.2) Calculate the fitness values  $h(x_i)$  for  $i = 1, \dots, n$ .
    (1.3) Determine the personal best positions  $p_i = x_i$  and the global position  $p_g$ .
2  Repeat until stopping criteria are satisfied.
    (2.1) Calculate particle velocity according Eq. (1).
    (2.2) Update particle position according Eq. (2).
    (2.3) Calculate the fitness values  $h(x_i)$ .
    (2.4) Update personal and global best positions  $p_i$  and  $p_g$ .
3  Output  $p_g = \text{argmin } h(x)$  with  $gbest = h(p_g)$ .

```

The initial velocity for each particle may be set equal to 0 or be randomly assigned from $U(0,1)$. A key difference in our PSO version is that our version requires that the weights in designs are positive and they sum to unity and we pull back particles that wander outside of the design space to the boundary. The rationale for having this feature is that many D -optimal designs have support points at the boundary of the design space, see [17] for example. Without this feature that we have introduced into the PSO algorithm, our experience is that PSO becomes either very slow or it fails to find the optimal design, especially in high dimensional problems.

3. Generating Optimal Designs for Biomedical Studies using PSO

In this section, we discuss the statistical background and apply PSO to find various types of optimal designs for common models in the biomedical studies. These models may appear small in terms of the number of parameters that they have but as noted in Konstantinou et al. [18], finding optimal designs for such models can still be problematic using traditional numerical methods or analytically.

Here and throughout, our focus is approximate designs, which are probability measures defined on the given design space X [19]. This means that once we are given a pre-determined sample size n , a given model and a design criterion (or objective function) $h(\cdot)$, our optimization problem is to find the number (k) of design points (or support points) required, the locations of the design points $x_1, \dots, x_k$ in X and the weight distribution $w_1, \dots, w_k$ at these points to optimize the given criterion $h(\cdot)$. These weights naturally sum to unity and all weights are strictly between 0 and 1. In practice, we round up each nw_i to a positive integer and take this number of observations at x_i , $i = 1, \dots, k$ subject to $nw_1 + \dots + nw_k = N$.

A common design problem is how to choose the value for k , along with the vectors values for $x_1, \dots, x_k$ and $w_1, \dots, w_k$ to optimize a given criterion or objective function.

Our statistical models have a known mean response function $\eta(x, \theta)$ apart from the values of the parameter vector θ and all have independent normal errors, each with mean zero and constant variance. The worth of a design ξ is measured by its Fisher information matrix $M(\xi, \theta)$ defined as the negative expectation of the second derivatives of its total log likelihood function with respect to the parameters.

The design criterion is formulated as a convex function of the information matrix. For example, for D -optimality we have $h(M(\xi, \theta)) = -\ln|M(\xi, \theta)|$ and for c -optimality, we have $h(M(\xi, \theta)) = c^T(\theta)M(\xi, \theta)^{-1}c(\theta)$ where $c(\theta)$ is a user-specified function of the model parameters to be estimated. Designs that minimize these criteria are D -optimal and c -optimal designs respectively and the minimization is over all designs on the given design space X . Because each design criterion is convex, we can use results from convex analysis and verify the optimality of an approximate design ξ by examining the plot of its directional derivative

of the criterion over the design space X . For example, if we let $\frac{\partial \eta(x, \theta)}{\partial \theta}$ be the gradient of $\eta(x, \theta)$ with respect to θ , an approximate design ξ^* is D -optimal over all approximate designs on X if and only if we have,

$$\frac{\partial \eta(x, \theta)^T}{\partial \theta} M(\xi^*, \theta)^{-1} \frac{\partial \eta(x, \theta)}{\partial \theta} - p \leq 0 \quad \forall x \in X,$$

with equality at the support points of ξ^* . In the design literature, the above checking conditions are frequently called equivalence theorems.

For low dimensional design space, such as when we have only a dose interval to decide what and how many doses to use and how many subjects to assign to each dose, the D -optimality of a design can be easily verified by plotting the graph of the function on the left hand side of the above inequality and examining it to determine whether conditions in the equivalence theorem are satisfied. If the candidate design satisfies all conditions in the equivalence theorem, the design is optimal; otherwise it is not. Figure 1 in subsection 3.2 is an example of such a plot, sometimes informally referred to as an equivalence plot. The graph satisfies the conditions in the equivalence theorem and so confirms local D -optimality of the design symmetrically supported at $-0.9217, -0.5921, 0.5921$ and 0.9217 for the quadratic logistic model. The weights at the points are not displayed and they are equal to $0.2966, 0.2034, 0.2034$ and 0.2966 respectively.

The PSO code is initiated as follows. The user first selects the flock size, the number of iterations desired and parameters for the design problem, such as the limits of the design space and the nominal values of the model parameters for the particular problem at hand. The cognitive and social learning parameters are all set to $\gamma_1 = \gamma_2 = 2$. Consequently, we only have to fuss with the flock size and the number of iterations. This simplifies the process and is an appealing feature of PSO. The search for the optimal design by PSO begins with a

randomly generated flock of size specified by the user. These designs all have the same fixed number of points, k equal to the number of parameters in the mean response function. When the algorithm is run, the k -point optimal design is usually found very quickly. When the design optimality criterion is convex, which is true for all our examples, the generated design can be verified using an equivalence theorem derived from the directional derivative.

If the above strategy fails to produce the optimal design, meaning that repeated searches by PSO with different number of iterations and different flock sizes produced a design that did not meet the equivalence theorem conditions, we continue the search to all designs with $k + 1$ points. Our experience is that such a strategy always produce a locally optimal design and we only need to search among designs with $k + j$ points where j is usually 1 or 2. On the other hand, if we have over-specified the number of support points required by the optimal design, then PSO will report an optimal design with some identical points or some points with extremely small weights. The masses at these identical points are then summed to obtain the optimal design. Such situations arise when the optimal design has a singular information matrix, as in subsection 3.1 where we wanted to find the best sampling times for estimating the time a drug takes to reach its maximum concentration or the length of time (area under curve) the drug resides in a targeted organ in a compartmental model.

We have set up mirror websites at 3 places with examples discussed in this paper so that the reader can download the P-codes, verify our results and appreciate how PSO works in practice. One site is housed at UCLA at <http://optimal-design.biostat.ucla.edu/podpack/> and the other two sites are housed in Taiwan at <http://www.math.ntu.edu.tw/~optdesign> and <http://www.stat.ncku.edu.tw/optdesign>. The instructions for downloading the MATLAB P-codes are available on websites. Many of the codes can be readily changed to find another type of optimal design for the same model or a different model. Typically, the only changes that are required are in the information matrix and the design criterion while leaving much of the rest of the code intact. For example, it took the first author about 15 minutes to re-code the PSO code for finding the c -optimal design for estimating the area under the curve in the compartmental model to one for estimating the two model parameters in the survival model, excluding of time to read the paper by Konstantinou et al. [18].

In what follows, we present different types of optimal designs found by PSO for the following models: (i) a 3-parameter compartmental models used in pharmacokinetics, (ii) 2 and 3-parameter logistic models for studying binary responses, (iii) a double exponential model for studying tumor regrowth rate and (iv) a 2-parameter survival model. The design problems involve D and c -optimality and were selected in part to compare PSO results and the theoretical results from the literature. We provide a bit more technical details in the first example, including an exemplary impact of choices of flock size and number of iterations on the performance of PSO for our problems. For each problem, we provide motivation and sample applications of the PSO-generated design to determine optimal sampling times in real problems.

All results reported in Tables 1-3 were obtained using a laptop equipped with an Intel i5 2520M, 2.5GHz, 8GB RAM, Windows 7 professional operating system and Matlab version 2010b.

3.1. Locally optimal designs for compartmental models

Compartmental models are commonly used to model drug movement through the body, see for example ([20, 21, 22, 23, 24, 25]). A typical model used frequently in pharmaceutical studies is the 3-compartmental model with mean response given by

$$\eta(x, \theta) = \theta_3 \{ \exp(-\theta_2 x) - \exp(-\theta_1 x) \}, \quad x > 0.$$

This model is not limited to pharmaceutical applications; for instance, Fresen [26] used the same model in veterinary science and modeled the effect of theophylline on horses. For space consideration, we focus on the 3-parameter model in this paper.

Let $\theta = (\theta_1, \theta_2, \theta_3)^T$ be the vector of model parameters for the 3-parameter compartmental model and let $\theta^0 = (\theta_1^0, \theta_2^0, \theta_3^0)^T$ be the vector of nominal values for θ for the compartmental model given by

$$\eta(x, \theta) \approx \eta(x, \theta^0) + \left[\frac{\partial \eta(x, \theta)}{\partial \theta} \Big|_{\theta^0} \right]^T (\theta - \theta^0)$$

with $\theta_2 > \theta_1 > 0$ and $\theta_3 > 0$. A direct calculation shows the gradient of the approximated mean function at the point x is

$$f^T(x, \theta^0) = \left(\frac{\partial \eta(x, \theta)}{\partial \theta_1}, \frac{\partial \eta(x, \theta)}{\partial \theta_2}, \frac{\partial \eta(x, \theta)}{\partial \theta_3} \right) \Big|_{\theta^0},$$

with

$$\begin{aligned} \frac{\partial \eta(x, \theta)}{\partial \theta_1} &= x \theta_3 \exp(-\theta_1 x) \\ \frac{\partial \eta(x, \theta)}{\partial \theta_2} &= -x \theta_3 \exp(-\theta_2 x) \\ \frac{\partial \eta(x, \theta)}{\partial \theta_3} &= \exp(-\theta_2 x) - \exp(-\theta_1 x) \end{aligned}$$

When the sample size is fixed and the design ξ takes independent observations at $x_1, \dots, x_n$,

the total information matrix is proportional to $M(\xi, \theta^0) = \sum_i f(x_i, \theta^0) f^T(x_i, \theta^0)$. If there are replications, we use weights $w_1, \dots, w_k$ to represent the proportions of observations at

these points and the information matrix becomes $\sum_i w_i f(x_i, \theta^0) f^T(x_i, \theta^0)$, apart from an unimportant multiplicative constant.

To implement PSO to find locally D and c -optimal design for the 3-parameter compartment model, we assumed for comparison purposes the same dose interval and the same set of

nominal parameters used in Atkinson, Donev and Tobias's book [27] with $\theta_1^0=0.05884$, $\theta_2^0=4.298$ and $\theta_3^0=21.8$. These values were obtained from the least square estimates of the parameters from Fresen [26], who implemented a design for the same model with 18 points. Specifically, his study involved six horses where each received 15mg/kg of theophylline. To estimate the model parameters, 18 measurements of the concentration of theophylline, as responses, were collected at different time points. Our research questions are why 18 measurements, how were the points selected, and whether this design is an efficient design or not?

We used PSO to find an optimal design for estimating all model parameters and use it to compare the efficiency of the implemented design. As a first attempt, we used 100 iterations and a flock size of 100 particles each with 3 support points in the PSO algorithm to find the locally D -optimal design for this compartmental model. This is a 5 dimension optimization problem to find the best choices of x_1, x_2, x_3, w_1, w_2 , where $0 < x_1 < x_2 < x_3 < 30$, $0 < w_1 < 1$, $0 < w_2 < 1$, $0 < w_3 = 1 - w_1 - w_2$. The PSO-generated design was equally supported at 0.2288, 1.3886 and 18.4168 and it coincides with the locally D -optimal design in Atkinson, Donev and Tobias's book [27] on page 264. The equivalence plot confirms the optimality of the design among all designs on the designated dose interval and we do not need to search further. The CPU time was 0.408 seconds. If we had used just 50 particles and 50 iterations, we would also have obtained the same design in 0.381 seconds.

There are three common characteristics of a drug are its maximum concentration in the targeted compartment, time to maximum concentration in the compartment and the average time it spends inside the compartment. We discussed only the latter two objectives for space consideration because finding optimal design for the first objective can be carried out in a similar way. The locally c -optimal design for estimating the time to max concentration of a drug can be found by integration directly from the model. This time as a function of the model parameters is

$$c(\theta) = \frac{\ln \theta_1 - \ln \theta_2}{\theta_1 - \theta_2}.$$

Our goal then is to choose a design to minimize the asymptotic variance of the estimated time ($\hat{\theta}$) given by

$$h(M(\xi, \theta)) = \left[\frac{\partial c(\theta)}{\partial \theta} \right]^T M(\xi, \theta)^{-1} \left[\frac{\partial c(\theta)}{\partial \theta} \right].$$

Here $\frac{\partial c(\theta)}{\partial \theta} = (a/\theta_1 - b)/a^2, (b - a/\theta_2)/a^2, 0)$ where $a = \theta_1 - \theta_2$, $b = \ln \theta_1 - \ln \theta_2$ and $M(\xi, \theta)^{-}$ is a generalized inverse of the information matrix. Using the same set of nominal values of the parameters, we use PSO to minimize the variance of $c(\hat{\theta})$ in a similar way as before. In our case here, we started with two points implying that we sought to minimize $h(M(\xi, \hat{\theta}))$ by choice of x_1, x_2, w_1 subject to $0 < x_1 < x_2 < 30$, $0 < w_1 < 1$ and $w_2 = 1 - w_1$.

The locally optimal design for estimating the time to maximum concentration is a c -optimal design with only 2 support points, which means that its information matrix is singular because the matrix is now a sum of two rank-one matrices. To get around having to find the inverse of singular information matrix $M(\xi, \theta)$, we followed convention and added a small multiple to the identity matrix and worked with the invertible matrix

$$M_{\epsilon}(\xi, \theta) = M(\xi, \theta) + \epsilon I_3.$$

We implemented PSO to find the optimal values of x_1, x_2, w_1 subject to $0 < x_1 < x_2 < 10$ and $0 < w_1 < 1$ with $w_2 = 1 - w_1$. The PSO parameters we used were $\epsilon = 10^{-6}$ and 200 particles all with $k = 2$ points. Expecting a singular information matrix for the optimal design, we allowed for a larger number of iterations and after 1000 iterations, PSO generated a two-point design supported at 0.1793 and 3.5658 with weight 0.3938 at the latter point. This optimal design also agrees with the c -optimal design reported in Atkinson, Donev and Tobias's book [27] on page 264.

A similar procedure is used to find the optimal design for estimating the time the drug spends inside the compartment. A direct calculation shows this function is simply $1/\theta_1 - 1/\theta_2$. Proceeding as before, we applied PSO to minimize the asymptotic variance of the estimated AUC. Setting $k = 2$ and using 100 particles and 1000 iterations, PSO took 6.185 seconds to generate a two-point design supported at 0.2326 and 17.6339 with mass 0.0135 at the smaller point. This design also agreed with the optimal design in Atkinson, Donev and Tobias's book [27] on page 264.

It is interesting to observe that had we used a flock of 200 particles all with $k = 3$ points and 500 iterations, the PSO-generated design was supported at 0.2337, 17.6269 and 17.7176 with mass distribution at these points equal to 0.0135, 0.8983 and 0.0882. Increasing the number of iterations to 1000 results in a design support at 0.2332, 17.6336 and 17.6626 with mass distribution at these points equal to 0.0135, 0.9535 and 0.03296. These results are consistent with the expectation that a longer iteration and/or more particles usually produces a higher quality solution. It also shows a very nice feature of PSO in that when we over-specify the number of support points the optimal design has, PSO can also automatically find the optimal design directly; in the above case, the 3 points found above get increasingly closer to 2 points as more iterations are used, leaving the mass at the smaller point unchanged.

This example shows that design issues are important and a properly constructed design can provide substantial cost savings. In Fresen's study, the 18 time points seemed excessive compared with the locally D -optimal design that requires only 3 points and at the same time provides the most accurate inference for the 3 parameters among all designs. If a measurement is costly to collect in terms of time and labor, the optimal design provides further benefits. In terms of statistical efficiencies, one can evaluate the efficiency of the Fresen's design by examining the ratio (or some function thereof) of the criterion values evaluated at the Fresen's design and the optimal design. For the D -optimality criterion and the criteria for estimating the AUC, time to maximum concentration and the maximum

concentration in the targeted compartment, a direct calculation shows these efficiencies are 67.61%, 24%, 28.6% and 36.77% respectively. The practical implications are that about 33% more resources are required for Fresen's design to do as well as the locally D -optimal design, about 75% more resources are required for Fresen's design to do as well as the locally AUC-optimal design, etc.

3.2. Locally Optimal Designs for the simple and quadratic logistic models

Binary responses are frequently modeled using linear logistic models because of their simplicity and ease of interpretation. Sometimes quadratic logistic models with three parameters are appropriate; see for example, Fan et al. (2002) [28], who used the quadratic logistic model to analyze a HIV model or a study by a microwave popcorn manufacturer who wanted to find out how much salt consumers like in their popcorn. Respondents rated the taste on a scale of 1=low to 5=high on popcorns made with 0,1,2 or 3 teaspoons of salts. There were 10 observations at each level of salt contents and the manufacturer's goal was to ascertain the maximum probability of a favorable response as a function of how much salt is added to the popcorn package. In this study described in JMP Software at www.jmp.com/support/help/Example_of_a_Quadratic_Ordinal_Logistic_Model.shtml#101245, it is natural to use a quadratic ordinal logistic model because of the expected nature of the trend of the rankings by respondents.

A further and common application of quadratic logistic models is given Bliss (1935) [29], where he analyzed binary responses from an experiment to study mortality of confused flour beetles, *Tribolium confusum*, given 8 different doses of gaseous carbon disulphide. Prescribed volumes of carbon disulphide were added to flasks in which a tabular cloth cage with a batch of about 30 beetles was suspended. For each concentration, duplicate batches of beetles were used and at the end of a 5-hour period, the proportion of beetles killed was noted along and the actual concentration of gaseous carbon disulphide in the flask, measured in mg/l, was determined by a volumetric analysis. When the logistic model was fitted to the proportions as a quadratic function of the dose concentrations, the improvement in the fit over the linear model reduces the model deviance significantly, suggesting that the quadratic model provides a much better fit than the simpler linear logistic model. Our research questions here and for the above problems were how the number of doses or teaspoons of salt was selected and how the concentrations were chosen for the study for estimating the model parameters. Specifically, are there designs with fewer doses at different doses that would provide better or the highest quality of inference for the simple or quadratic logistic model than the implemented designs?

Locally D -optimal designs for estimating all the model parameters provides the best inference. Probably the first description of the locally D -optimal design for the simple logistic model was given in a doctoral thesis [30] for the prototype interval $[-1, 1]$ and reported in Silvey [8]. The formula is complicated for a relatively simple model. When we ran PSO to compare results, we were unable to verify Ford's results. A corrected formula for the locally D -optimal design on an arbitrary interval was given in Sebastiani and Settimi [31] and we were able to verify their results using the MATLAB P-code available on the website.

Quadratic logistic models are sometimes employed to explore possible curvature in the model or for estimating an interesting characteristic of an agent in a dose response study. For example, in radiology and radiotherapy there is often interest in estimating the ratio of the coefficients associated with the linear and quadratic terms in the quadratic logistic model [32]. Selected locally c -optimal designs for the quadratic logistic model were found theoretically in Fornius and Nyquist [33] after re-parameterizing the model in the following way using their notation:

$$\ln \frac{Ey}{1 - Ey} = \alpha + \beta(x - \mu)^2.$$

Here y is the binary response taking on values 0 or 1 with probabilities that vary with the dose levels x . We provide on our website PSO codes for generating locally D -optimal design on an arbitrary interval. For example, suppose the design interval is $[-3, 1]$ and the nominal values for the 3 parameters are $\alpha = 2$, $\beta = 3$ and $\mu = 0$ and we began our search among all 3-point designs. With a flock size of 128 and the number of iterations set at 150, PSO took 6.623 seconds to find a design equally supported at -0.7270, 0 and 0.7270. The equivalence plot confirms the D -optimality of this design. If fewer number of iterations were used, say 50 iterations, the design points for the optimal design will also emerge quickly, except that the weights are only approximately equal. PSO took 3.104 seconds to produce the design and reports the design has an efficiency of 99.98%.

Interestingly, when the maximum probability of response is high, there are 4-point locally D -optimal designs. For instance suppose the design interval is $[-1, 1]$ and the nominal values for the 3 parameters are $\alpha = 3$, $\beta = -5$ and $\mu = 0$. With a flock size of 256 and the number of iterations set at 200, PSO took 18.317 seconds to find a symmetric design supported at -0.9217, -0.5921, 0.5921 and 0.9217. The weights at -0.9217 and at -0.5921 are 0.2966 and 0.2034, respectively. Figure 1 is generated from the P-code from our websites and shows the equivalence plot of this PSO-generated 4-point design for the quadratic logistic model. The plot is bounded above by 0 throughout the scaled design interval $[-1, 1]$ with equality at the support points of the PSO-generated design and so the figure confirms its D -optimality. Table 1 displays locally D -optimal designs for estimating the three parameters in the quadratic logistic regression model for various nominal values of the parameters and on different design intervals.

As always here and elsewhere, in order to ensure a higher chance that PSO will generate the optimal design, we report flock size and number of iterations larger than are usually necessary. Frequently, smaller flock size and smaller number of iterations will suffice which mean shorter CPU time can usually also produce the optimal design.

There are additional interesting design questions for the quadratic model that a c -optimal can be useful. Quadratic logistic models are sometimes employed to explore possible curvature in the model or for estimating an interesting characteristic of an agent in a dose response study. In the former case estimating the coefficient associated with the quadratic term provides an indication of curvature presence in the the model. An example of an interesting

quantity to estimate in radiology and radiotherapy is the ratio of the coefficients associated with the linear and quadratic terms in the quadratic logistic model [32]. PSO codes can also be implemented directly to find these c -optimal designs.

3.3. Locally optimal designs for a double exponential model

Double-exponential regrowth model was developed by Demidenko [34] to describe the dynamics of post-irradiated tumors based on the two-compartment model. One may categorize tumor cells as proliferating or quiescent and under appropriate assumptions, the natural logarithm of the tumor volume of the two kinds of cells at time x may be expressed as

$$y = \alpha + \ln \left[\beta e^{\nu x} + (1 - \beta) e^{-\phi x} \right] + \varepsilon,$$

where $\varepsilon \sim N(0, \sigma^2)$ and α is the logarithm of the initial tumor volume at time $x = 0$. The design problem of interest here is to judiciously select the number of measurements and sampling time points from a given time interval to estimate the volume of the tumor size as accurately as possible.

Let $\theta = (\alpha, \beta, \nu, \phi)^T$ be the vector of model parameters and its nominal values be θ^0 . To construct the locally D -optimal design by maximizing $|M(\xi, \theta)|$, we first verify the gradient of the mean function is

$$f(t, \theta) = \left(1, \frac{e^{\nu t} - e^{-\phi t}}{\beta e^{\nu t} + (1 - \beta) e^{-\phi t}}, \frac{\beta x e^{\nu x}}{\beta e^{\nu x} + (1 - \beta) e^{-\phi x}} - \frac{-(1 - \beta) x e^{-\phi x}}{\beta e^{\nu x} + (1 - \beta) e^{-\phi x}} \right)^T$$

and then linearize the mean function of the logarithm tumor size by a first order Taylor's expansion at θ^0 to obtain the information matrix.

Li and Balakrishnan [17] showed that $|M(\xi, \theta)|$ depends only on β^0 and $\nu^0 + \phi^0$, which implies that to generate the locally D -optimal design, only a nominal value for β and a nominal value for the sum of the two parameters ν and ϕ are required. Proceeding as before and using PSO codes downloadable from our websites, one can verify that PSO readily generated locally D -optimal designs that matched those in Li and Balakrishnan [17]. For instance, suppose we set $\beta = 0.2$, $\nu + \phi = 0.4$ and the time interval is $[0, 10]$. With 100 particles and 100 iterations, PSO produced in 1.041 seconds the locally D -optimal design equally supported at 0, 2.660, 6.707 and 10 reported on the first line in Table 1 in [17]. Likewise, the same flock size and same number of iterations also produced in 1.085 seconds the locally D -optimal design in the last row of their Table 1 where $\beta = 0.8$ and $\nu + \phi = 1$. Other c -optimal designs in their paper can be similarly verified.

3.4. Locally optimal designs for a survival model

Konstantinou et al. [18] investigated a two-parameter exponential model with type I right censored data, where all individuals entered the study at the same time and stayed until a

user-specified time c or until failure, whichever was earlier. Right-censoring occurs when survival times are greater than c . Let $t_1, \dots, t_n$ be the observed values for n subjects and $x_i \in X$ be the experimental condition at which the i th observation was taken. The design space could be a dose interval or the set $\{0,1\}$ representing two treatment conditions, say treated or not. The maximum time for observing outcomes in the study is $c = 30$ so that all observations are right censored if the outcome is not observed by that time. They used an exponential regression model with probability density function $g(t_i)$ and a survival function $S(t_i)$ given respectively by

$$g(t_i) = e^{\alpha + \beta x_i} \exp(-t_i e^{\alpha + \beta x_i})$$

and

$$S(t_i) = \exp(-t_i e^{\alpha + \beta x_i}).$$

Without loss of generality we assumed that the first k observations were failure times and rest $n - k$ observations were right censored. A direct calculation shows that when all observations (x_i, t_i) , $i = 1, \dots, n$ are independent, the log-likelihood is

$$l(\alpha, \beta, x_1, \dots, x_n) = \ln \left\{ \prod_{i=1}^k g(t_i) \prod_{i=k+1}^n S(t_i) \right\} = \sum_{i=1}^k (\alpha + \beta x_i) - \sum_{i=k+1}^n t_i \exp(\alpha + \beta x_i).$$

It follows that the information matrix using design ξ is

$$M(\xi, \alpha, \beta) = \sum_{i=1}^n \left(1 - \exp(-c e^{\alpha + \beta x_i}) \right) \begin{bmatrix} 1 & x_i \\ x_i & x_i^2 \end{bmatrix}.$$

The information matrix $M(\xi, \theta)$ depends on unknown parameters α and β because the model is nonlinear. Four sets of nominal value were used to find the locally D -optimal design, which is always equally supported at two points. Using 20 particles and 50 iterations, PSO was able to find the locally D -optimal designs as claimed in their paper.

In practice, the parameter β in the model always has a clear biological interpretations and so it is often of interest. If we have a dose response study β measures the effect of increasing dose on the response and if we have two treatment conditions, β represents the effect on the hazard of death when say the new treatment is compared with the placebo condition. An appropriate design to use here for estimating β is the locally c -optimal design that gives the smallest asymptotic variance of the estimate. This design is the same as the optimal design for minimizing the criterion

$$\begin{bmatrix} 0 & 1 \end{bmatrix} M^{-}(\xi, \alpha, \beta) \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

For this two-parameter model, it can be shown that the c -optimal design is always supported at 0 and 1 but with unequal weights that depend on the nominal values. Using 20 particles and 50 iterations, PSO was able to find and verify all the c -optimal designs for the four sets of nominal values in Konstantinou et al. [18] and these weights are shown in Table 2.

4. Discussion

We discussed using PSO to find locally D and c -optimal designs for compartmental models, logistic models, a double exponential model useful for monitoring tumor regrowth and estimating parameters in a survival model. The computational experience we had with these and many other problems we had looked at were similar to what is reported in the literature. First, many parameters in the PSO did not seem to matter much. Following common usage in the literature, see for example, [35, 36], we set $\gamma_1 = \gamma_2 = 2$ and they all seemed to work well in the problems discussed here; other values tended to take a longer time for the swarm to converge, if it did at all. The only two parameters that we changed from problem to problem were number of iterations and the flock size.

In general, PSO generates the optimal designs very quickly for all our examples in this paper and elsewhere compared with our experiences with other algorithms. The CPU time for generating each optimal design each time for the same setting varies slightly because of the stochastic nature of the algorithm. In each instance, we verified that the generated designs were optimal up to the 4 decimal places for both the support points and the weights using an equivalence theorem. Table 3 below reports the CPU times and their standard deviations required by PSO to find the various optimal designs using different particle numbers when we averaged over 30 replications. Convergence for the PSO was defined as when the design criterion value does not change by more than 10^{-7} deviation from the known optimum value.

From Table 3, we observe that on average, less than 0.3 second in CPU time was required to find the various locally D -optimal designs and a longer time is required to generate c -optimal designs with a singular information matrix. In the latter case, CPU time required is generally still short and only required less than 10 seconds to find the locally c -optimal designs for estimating time to maximum concentration or the area under the curve in the 3-parameter compartmental model. Not surprisingly, a larger number of particles to cover a larger area of search space to begin with will require a longer time to find the optimal designs, but the increase in time is quite negligible for all our examples. We note that the standard deviations of the CPU times for PSO to generate the design with 20 particles are usually larger than those when a large number of particles is used. This is because the smaller number of starting designs (particles) were not enough to cover the design space adequately and in a consistent manner, and so more variability in the search capability.

Our experience is that in terms of time, PSO typically identifies the optimum usually in a few seconds of CPU time for most of the problems we had worked with. They include different types of optimal designs for nonlinear models with up to 5 parameters or models with rational polynomials as mean functions. Additionally, we have applied PSO to solve minimax design problems which are notoriously difficult to solve because they involve two layers of optimization over two spaces. If one of these spaces is discrete, as in *E*-optimal design problems or minimax single-parameter optimal design problems, time required to determine the optimal design is significantly shorter than when the two spaces are non-discrete, as in finding a design to minimize the maximum variance of the predicted response over a compact design interval. For such minimax problems, where optimization is sought over two non-discrete compact spaces, we are not aware of any traditional algorithm that converge to the minimax optimal design even for linear models.

Our boldest attempt to date was to test PSO search ability for finding the *D*-optimal design for the Scheffe's quadratic mixture model with 10 factors. This optimization problem involved optimizing hundreds of variables and we were pleased that PSO was able to get very close to the optimal design and found a design with a *D*-efficiency of 99.98%.

How does PSO compare with other metaheuristic algorithms, such as differential evolution (DE) algorithm proposed in [37]? We choose to compare PSO with DE because DE is typically viewed as a competitor to PSO and DE seems to be a popular optimization tool in engineering applications. For this purpose, we used the DE default program in [37], the authors claimed that DE, based on multiple testing functions, outperformed other popular metaheuristic algorithms including Adaptive Simulated Annealing (ASA), the Annealed Nelder and Mead approach (ANM), the Breeder Genetic Algorithm (BGA), The EASY Evolution Strategy and the method of Stochastic Differential Equations.

Comparing algorithms that use different strategies and with different numbers and sets of tuning parameters can be difficult because one may argue that tuning parameters were not properly selected. Our experience is that for the types of problems we want to optimize, only two tuning parameters in the two algorithms seem to matter, a finding not too unlike others reported in the literature. For the comparison here, we did preliminary runs with various tuning parameters to first estimate the minimal flock size and iteration number that would generate the optimal design with 90% chance over 10 replicates. We informally call this the 90% rule. If a particular tuning parameter setting fails, we repeat the search by increasing the flock size by 20 and the iteration numbers by 100 sequentially.

The key tuning parameters in DE are the crossover constant $CR \in [0, 1]$ and the amplification factor $F \in [0, 2]$. For our problems, we used the default values $CR = 0.7$ and $F = 0.8$ provided by the authors. Table 4 reports the CPU time for finding the optimal designs by the two algorithms, averaged over 10 replicates. In each instance, the generated designs were verified to be optimal up to the 4 decimal places for both the support points and the weights. In all cases, PSO outperforms DE and sometimes by as much as 41 ($= 57.4/1.4$) times in terms of CPU times as in the case for finding *D*-optimal design for the quadratic logistic model. We also compared the performance of the two algorithms when we have higher dimensional problems when we sought a locally *D*-optimal design for a Poisson

model with 3 variables and all pairwise interaction terms plus an intercept. This problem has 7 parameters and a search among all 7-point designs for the optimum is tantamount to solving a 27-dimensional constrained optimization problem. For this problem, we used the 90% rule mentioned above and employed 400 iterations and a flock size of 60 for PSO and 40 for DE. The CPU time required by PSO was 2.8 seconds and the CPU time required by DE was 51.3, showing that PSO again outperforms DE. The consistently faster performance of PSO over DE we observed here suggests PSO works well for such problems but we make no claim about the general superiority of PSO over DE for solving optimization problems.

We note that there have been many proposed changes in the PSO algorithm since it was proposed about two decades ago and it is not clear now what the standard or basic PSO version refers to. Some of the proposed changes in the PSO algorithm include not working with the global best but instead with ring and local topologies where the global best is now replaced by an optimum as determined by a couple of neighboring particles to speed up convergence, see for example, Bratton and Kennedy [38] and Mendes [39]. On the particle swarm central website <http://particleswarm.info>, there are also versions of the PSO algorithms in 2006, 2007 and 2008 that used quasi-random approaches rather than random searches. Other proposed changes include Helwig (2010) [40], who proposed confinement methods that bring particles back to the boundaries of the search space and Miranda et al. (2008) [41], who proposed adaptive searches based on PSO.

The version of PSO presented in this paper is a very basic one and we demonstrated that despite its simplicity, it still works remarkably well for all our problems. We also briefly compared our version with one that uses local topologies and found the latter underperformed for the problems we looked at. Figure 2 shows the convergence rate to the locally D -optimal design for the compartmental model in subsection 3.1, and Figure 3 is the corresponding graph for the locally c -optimal design for estimating the AUC of the compartmental model. In each graph, the convergence rate is expressed as a function of number of iterations and $G_{\text{best}}(L_{\text{best}})$ denotes the plot obtained using the global(local) best topology. For the local topology, we employed the simplest ring model where each particle is connected only to its adjacent two particles. For both design problems, we used 100 particles and 500 iterations in our PSO to optimize the objective functions. Results were averaged over 20 independent runs and they showed the version that used local topologies took a uniformly longer time to reach the optimum value than our version using the best global search strategy. This shows that depending on the problem at hand, the most basic version of PSO can still perform very well and sometimes even outperform recent versions of PSO. When the basic PSO version used in this paper fails to find the optimum repeatedly, the user may want to consider using newer versions of PSO to find the optimum. Some examples of recent versions of PSO are [42, 43, 44, 45] with suggestions for different choices for the tuning parameters.

5. Conclusion

Particle swarm optimization is a powerful, flexible and interesting tool for solving optimization problems but appears greatly under-utilized in the statistical literature. We have shown here that PSO is an efficient and novel way for finding optimal experimental designs

in statistics. Our applications were in biostatistical problems but clearly they can be applied to solve general optimization problems in statistics. An advantage of PSO is that it is a versatile algorithm in that its success to find the optimum is generally not dependent on the technical requirements imposed on the problem. For example, Li and Balakrishnan [17] and Konstantinou et al. [18] assumed technical conditions to arrive at the theoretical descriptions of the optimal designs. PSO can find optimal designs that satisfy the technical conditions and also able to find optimal designs when the conditions do not apply, suggesting that PSO can find optimal designs for a wider class of problems [46, 47, 48, 49].

We close with two important notes. First, the only role that the convexity assumption in our design criterion plays is that it enables us to definitively confirm the optimality of the PSO-generated design via the equivalence theorem. If it is not optimal, the convexity of the design criterion also provides an assessment of how far the PSO-generated design is from the optimum via a lower bound of the efficiency of the generated design (Pazman, 1986) [50]. Second, we have to date also applied PSO and successfully found optimal designs under a convex criterion for more complicated models, including nonlinear models with 6 parameters, mixture experiments with 8 factors when the design space is a multi-dimensional irregular simplex and several types of minimax optimal designs when the criterion is still convex but not differentiable (Chen, et al., 2014) [51]. Our repeated successes with PSO this far have now encouraged us to further apply PSO to find optimal designs under non-convex criteria, where there is now no definitive and general way to check whether the generated design is optimal. Some examples are design criteria for finding exact optimal designs, minimum bias designs and designs that minimize the mean squared error. We hope to report further results in the future.

Acknowledgments

The research of Chen was partially supported by the National Science Council under grant NSC 101-2118-M-006-002-MY2 and the Mathematics Division of the National Center for Theoretical Sciences (South) in Taiwan. The research of Wang was partially supported by the National Science Council, the Taida Institute of Mathematical Sciences, and the National Center for Theoretical Sciences (Taipei Office). The idea for this manuscript originated at the Isaac Newton Institute for Mathematical Sciences at Cambridge, England when Wong was a visiting fellow there and a scientific adviser for a six month workshop in the design and analysis of experiment at the Institute. He would like to thank the Institute for the support during his repeated visits there in the second half of 2011. The research of Wong was partially supported by NSC, Mathematics Research Promotion Center with grant number 102-51 and the Ministry of Education, Taiwan, R.O.C. The Aim for the Top University Project to the National Cheng Kung University (NCKU). He wishes to thank National Taiwan University and National Cheng-Kung University for the hospitality during his repeated visits there to complete this project. In addition, the joint research of Chen, Wang and Wong reported in this paper was partially supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award Number R01GM107639. The content of the paper is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

All authors would also like to thank the referees for their valuable and helpful comments and to the editor for timely handling of the paper.

7. Appendix

It is instructive to display the objective function $h(\xi, \theta)$ that we wanted to optimize in each of the above problems. When this is possible, closed form expressions found from Mathematica are given below. For others, the Simplify command in Mathematica was not able to simplify the objective function that was complicated enough to require at least 4

pages of text output. We recall that θ is the vector of model parameters and for locally optimal designs, we assumed this vector is assumed known using nominal values θ^0 from previous or similar studies. The variables to optimize are the the number of points required, the locations of the points and the weight distribution of the design ξ at these points. For example, for a 3-point locally D -optimal design for the compartmental model defined on the positive real line in subsection 3.1, the objective function was to maximize $h(\xi, \theta) = \ln|M(\xi, \theta)|$, $\theta^T = (\theta_1, \theta_2, \theta_3)$ and the variables in the design ξ that we had to optimize were its support points x_1, x_2 and x_3 along with their corresponding weights w_1, w_2 and w_3 subject to the constraints that each weight is between 0 and 1 and they sum to unity.

$$h(M(\xi, \theta)) = \theta_3^4 w_1 w_2 w_3 \exp(-2(\theta_1 + \theta_2)(x_1 + x_2 + x_3)) \\ (-\exp(\theta_2 x_2 + \theta_1(x_1 + x_3))x_2(x_1 - x_3) - \exp(\theta_2 x_2 + \theta_2(x_1 + x_3))x_2(x_1 - x_3) \\ + \exp(\theta_2 x_1 + \theta_1(x_2 + x_3))x_1(x_2 - x_3) + \exp(\theta_1 x_1 + \theta_2(x_2 + x_3))x_1(x_2 - x_3) \\ + \exp(\theta_1 x_3 + \theta_2(x_1 + x_2))x_3(x_1 - x_2) + \exp(\theta_2 x_3 + \theta_1(x_1 + x_2))x_3(x_1 - x_2))^2.$$

The 4-point locally D -optimal design for the quadratic logistic model in subsection 3.2 was obtained by maximizing $h(\xi, \theta)$ shown below with $\theta^T = (\alpha, \beta, \mu)$. The variables in the design ξ that we wanted to optimize were the x_i 's and w_i 's. The x_i 's may represent the possible dose levels and the w_i 's are weights representing proportions of subjects to be assigned at the 4 chosen dose levels. The proportions w_i 's have the same set of constraints as above.

$$(M(\xi, \theta)) = 4\beta^2 \exp(3\alpha + \beta(\mu - x_1)^2 + \beta(\mu - x_2)^2) w_1 w_2 (x_1 - x_2)^2 (x_1 - x_3) (x_2 - x_3) \\ (\exp(\beta(\mu - x_3)^2) w_3 (x_1 - x_3) + \exp(2\alpha + \beta(\mu - x_3)^2) + \\ 2\beta(\mu - x_4)^2 (x_1 - x_3) (x_2 - x_3) + 2\exp(\alpha + \beta(\mu - x_3)^2 + \beta(\mu - x_4)^2) \\ (w_3 (x_1 - x_3) (x_2 - x_3) + w_4 (x_1 - x_4) (x_2 - x_4)) + \\ \exp(2\alpha + 2\beta(\mu - x_3)^2 + \beta(\mu - x_4)^2) w_4 (x_1 - x_4) (x_2 - x_4) + \\ \exp(\beta(\mu - x_4)^2) w_4 (x_1 - x_4) (x_2 - x_4)) / \\ ((1 + \exp(\alpha + \beta(\mu - x_1)^2)) (1 + \exp(\alpha + \beta(\mu - x_2)^2)) \\ (1 + \exp(\alpha + \beta(\mu - x_3)^2)) (1 + \exp(\alpha + \beta(\mu - x_4)^2))^2).$$

In each of the above problems, we first found the best design as determined by the optimal choices for the x_i 's and w_i 's within a subset of all k -point designs on the given design space and k was user-specified. In the first instance, we found the best 3-point design for the compartmental model and in the second case, we found the best 4-point design for the quadratic logistic model. Whether these designs remain optimal or not among all designs on the given design interval will have to be checked via the equivalence theorem. The theorem considers the directional derivative of the criterion evaluated at the design and confirms it is the best design among ALL designs on the design space.

References

1. Atkinson AC. The usefulness of optimum experimental designs. Journal of Royal Statistical Society B. 1996; 58:59–76.

2. Dette H, Beidermann S. Robust and Efficient Designs for the Michaelis-Menten Model. *Journal of the American Statistical Association*. 2003; 98:679–686.
3. Dette H, Melas VB, Wong WK. Optimal designs for goodness of fit of the Michaelis-Menten enzyme kinetic function. *Journal of the American Statistical Association*. 2005; 100:1370–1381.
4. Woods DC, Lewis SM, Eccleston JA, Russell KG. Designs for generalized linear models with several variables and model uncertainty. *Technometrics*. 2006; 48:284–292.
5. Lopez-Fidalgo J, Rivas-Lopez M, Del Campo R. Optimal designs for Cox regression. *Statistica Neerlandica*. 2009; 63:135–148.
6. Gilmour SG, Trinca LA. Bayesian *L*-optimal exact design of experiments for biological kinetic models. *Applied Statistics*. 2011; 61:237–251.
7. Berger, MPF.; Wong, WK. *An Introduction to Optimal Designs for Social and Biomedical Research*. John Wiley & Sons; Chichester: 2009.
8. Silvey, SD. *Optimal Design*. Chapman and Hall; London: 1980.
9. Fedorov, VV. *Theory of Optimal Experiments*. Academic Press; New York: 1972.
10. Wynn HP. Results in the theory and construction of *D*-optimum experimental designs. *Journal of Royal Statistical Society B*. 1972; 34:133–147.
11. Whitacre JM. Recent trends indicate rapid growth of nature-inspired optimization in academia and industry. *Computing*. 2011; 93:121–133.
12. Whitacre JM. Survival of the flexible: explaining the recent dominance of nature-inspired optimization within a rapidly evolving world. *Computing*. 2011; 93:135–146.
13. Garnier S, Gautrais J, Thraulaz G. The biological principles of swarm intelligence. *Swarm Intell*. 2007; 1:3–31.
14. Poli R, Kennedy J, Blackwell T. Particle swarm optimization: an overview. *Swarm Intell*. 2007; 1:33–57.
15. Yang, XS. *Nature-Inspired Metaheuristic Algorithms*. 2nd edition. Luniver Press; Frome: 2010.
16. Eberhart RC, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization. *Proceedings of the IEEE Congress Evolutionary Computation*. 2000; 1:84–88.
17. Li G, Balakrishnan N. Optimal designs for tumor regrowth models. *Journal of Statistical Planning and Inference*. 2011; 141:644–654.
18. Konstantinou, M.; Biedermann, S.; Kimber, A. *Optimal designs for two-parameter nonlinear models with application to survival models*. Southampton Statistical Research Institute, University of Southampton; 2011.
19. Kiefer, J. Jack Carl Kiefer Collected Papers III: Design of Experiments. Brown, LD.; Olkin, I.; Sacks, J.; Wynn, HP., editors. Springer-Verlag: 1980.
20. Kalicka R, Bochen D. Properties of D-optimal sampling schedule for compartmental models. *Biocybernetics and Biomedical Engineering*. 2005; 25:23–36.
21. Ogungbenro K, Dokoumetzidis A, Aarons L. Applications of optimal design methodologies in clinical pharmacology experiments. *Pharmaceutical Statistics*. 2009; 8:239–252. [PubMed: 19009585]
22. Lopez-Fidalgo J, Rodriguez-Diaz JM, Sanchez G, Santos-Martin MT. Optimal designs for compartmental models with correlated observations. *Journal of Applied Statistics*. 2005; 32:1075–1088.
23. Hooker AC, Foracchia M, Dodds MG, Vicini P. An evaluation of population D-optimal designs via pharmacokinetic simulations. *Annals of Biomedical Engineering*. 2003; 31:98–111. [PubMed: 12572660]
24. Jansen KM, Duffull SB, Tarning J, Lindegarch N, White NJ, Simpson JA. Optimal designs for population pharmacokinetic studies of the partner drugs co-administered with artemisinin derivatives in patients with complicated falciparum malaria. *Malaria Journal*. 2012; 11:143. [PubMed: 22551019]
25. Flegg JA, Gurin PJ, Nosten F, Ashley EA, Phyto AP, Dondorp AM, Fairhurst RM, Socheat D, Borrmann S, Bjorkman A, Martensson A, Mayxay M, Newton PN, Bethell D, Se Y, Noedl H, Diakite M, Djimde AA, Hien TT, White NJ, Stepniewska K. Optimal sampling designs for

- estimation of *Plasmodium falciparum* clearance rates in patients treated with artemisinin derivatives. *Malaria Journal*. 2013; 12:411. [PubMed: 24225303]
26. Fresen, J. M Sc Thesis. Department of Mathematical Statistics, University of Cape Town; 1984. Aspects of bioavailability studies.
 27. Atkinson, AC.; Donev, AN.; Tobias, RD. Optimum Experimental Designs, with SAS. Oxford University Press; USA: 2007.
 28. Fan X, Brauner CM, Wittkop L. Mathematical analysis of a HIV model with quadratic logistic growth term. *Discrete and Continuous Dynamical Systems - Series B*. 2012; 17:2359–2385.
 29. Bliss CI. The calculation of the dose-mortality curve. *Annals of Applied Biology*. 1935; 22:134–167.
 30. Ford, I. Ian Ford PhD Thesis. University of Glasgow; Scotland: 1976.
 31. Sebastiani P, Settimi R. A note on *D*-optimal designs for a logistic regression model. *Journal of Statistical Planning and Inference*. 1997; 59:359–368.
 32. Taylor JMG. The design of in vivo multifraction experiments to estimate the $\alpha - \beta$ ratio. *Radiation Research*. 1990; 121:91–97. [PubMed: 2300673]
 33. Fornius EF, Nyquist H. Using the canonical design space to obtain *c*-optimal designs for the quadratic logistic model. *Comm in Statistics-Theory and Methods*. 2010; 39:144–157.
 34. Demidenko E. The assessment of tumor response to treatment. *Applied Statistics*. 2006; 55:365–377.
 35. Rahmat-Samii Y, Jing N. Particle swarm optimization (PSO) in engineering electromagnetics: a nature-inspired evolutionary algorithm. *IEEE Journal*. 2007; 2013:1–8.
 36. Kim S, Li L. A novel global search algorithm for nonlinear mixed-effects models using particle swarm optimization. *J Pharmacokinetics and Pharmacodynamics*. 2011; 38:471–495.
 37. Storn R, Price K. Differential Evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. 1997; 11:341–359.
 38. Bratton D, Kennedy J. Defining a standard for particle swarm optimization. *IEEE Swarm Intelligence Symposium*. 2007:120–127.
 39. Mendes, R. PhD Thesis. Universidade do Minho; 2004. Population Topologies and Their Influence in Particle Swarm Performance.
 40. Helwig S. Particle Swarms for Constrained Optimization. Der Technischen Fakultät der Universität Erlangen-Nürnberg zur Erlangung des Grades. 2010
 41. Miranda V, Keko H, Duque AJ. Stochastic Star Communication Topology in Evolutionary Particle Swarms (EPSO). *International Journal of Computational Intelligent Research*. 2008:105–116.
 42. Kaucic M. A multi-start opposition-based particle swarm optimization algorithm with adaptive velocity for bound constrained global optimization. *Journal of Global Optimization*. 2013; 55:165–188.
 43. He J, Guo H. A modified particle swarm optimization algorithm. *Telkomnika*. 2013; 11:6209–6215.
 44. Ni Q, Deng J. Two improvement strategies for logistic dynamic particle swarm optimization in Adaptive and Natural Computing Algorithms. Springer. 2011:320–329.
 45. Ni Q, Deng J. A new logistic dynamic particle swarm optimization algorithm based on random topology. *The Scientific World Journal*. 2013:1–8.
 46. Marschner IC. Optimal design of clinical trials comparing several treatments with a control. *Pharmaceutical Statistics*. 2007; 6:33–33.
 47. Ogungbenro K, Dokoumetzidis A, Aarons L. Application of optimal design methodologies in clinical pharmacology experiments. *Pharmaceutical Statistics*. 2009; 8:239–252. [PubMed: 19009585]
 48. Vajjah R, Duffull SB. A generalisation of T-optimality for discriminating between competing models with an application to pharmacokinetic studies. *Pharmaceutical Statistics*. 2012; 11:503–510. [PubMed: 23059829]
 49. Biswas A, Lopez-Fidalgo J. Compound designs for dose-finding in the presence of nondesignable covariates. *Pharmaceutical Statistics*. 2013; 12:92–101. [PubMed: 23441044]
 50. Pazman, A. Foundations of Optimum Experimental Design. D Reidel Publishing Company; 1986.

51. Chen RB, Chang SP, Wang W, Tung HC, Wong WK. Minimax Optimal Designs via Particle Swarm Optimization Methods. *Statistics and Computing*. 2014 In press.

Figure 1.
The equivalence plot of the 4-point PSO-generated design for estimating the 3 parameters in the quadratic logistic model when $(\alpha^0, \beta^0, \mu^0) = (3, -5, 0)$.

Figure 2. D-optimality criterion value for the compartment model in subsection 3.1 using local topology or best optimum strategy versus iteration number.

Figure 3.
c-optimality criterion value for estimating the AUC in the compartment model in subsection 3.1 using local topology or best optimum strategy versus iteration number.

Table 1

Locally D -optimal designs for estimating the the three parameters in the quadratic logistic model for different nominal values and different design intervals.

$(\alpha^0, \beta^0, \mu^0)$	Design space	Locally D -optimal designs
$(0, -1, 0)$	$[-1, 1]$	$\begin{pmatrix} -1.000 & 0.0000 & 1.0000 \\ 0.3333 & 0.3333 & 0.3333 \end{pmatrix}$
$(0, -1, 0)$	$[-2, 2]$	$\begin{pmatrix} -1.4073 & 0.0000 & 1.4073 \\ 0.3333 & 0.3333 & 0.3333 \end{pmatrix}$
$(3, -1, 0)$	$[-1, 1]$	$\begin{pmatrix} -1.0000 & 0.0000 & 1.0000 \\ 0.3333 & 0.3333 & 0.3333 \end{pmatrix}$
$(3, -1, 0)$	$[-2, 2]$	$\begin{pmatrix} -2.0000 & -1.2506 & 1.2506 & 2.0000 \\ 0.3061 & 0.1939 & 0.1939 & 0.3061 \end{pmatrix}$
$(3, -1, 0)$	$[-4, 4]$	$\begin{pmatrix} -2.0609 & -1.3239 & 1.3239 & 2.0609 \\ 0.2966 & 0.2034 & 0.2034 & 0.2966 \end{pmatrix}$

Table 2Weights of selected locally c -optimal designs for the Survival Model.

Nominal values	w_1	w_2
$\alpha^0 = -2.163, \beta^0 = -0.1$	0.498	0.502
$\alpha^0 = -2.163, \beta^0 = -0.405$	0.491	0.509
$\alpha^0 = -2.163, \beta^0 = -1.526$	0.425	0.575
$\alpha^0 = -2.163, \beta^0 = -2.623$	0.324	0.676

Table 3

CPU times in seconds (standard deviations) required by PSO to generate the various locally optimal designs averaged over 30 replications.

Particle number	<i>D</i> -optimal design for compartmental model	AUC <i>c</i> -optimal design for compartmental model	Time-To-Max Con <i>c</i> -optimal design for compartmental model
20	1.477 (4.005)	4.514 (2.455)	6.511 (2.871)
40	0.131 (0.027)	4.796 (3.822)	6.012 (1.808)
60	0.146 (0.032)	6.026 (1.346)	6.674 (2.064)
80	0.160 (0.022)	6.137 (1.867)	7.427 (2.089)
100	0.179 (0.018)	6.312 (1.548)	8.565 (2.195)
120	0.193 (0.020)	6.295 (1.924)	9.044 (2.711)
140	0.225 (0.026)	7.513 (1.367)	9.302 (2.236)
160	0.240 (0.031)	6.608 (2.362)	9.273 (2.530)
180	0.251 (0.029)	7.609 (2.730)	10.233 (3.325)
200	0.275 (0.020)	8.741 (2.519)	9.708 (3.082)

Particle number	<i>D</i> -optimal design for double exp. model	<i>D</i> -optimal design for survival model	<i>c</i> -optimal design for survival model
20	0.773 (2.117)	0.042 (0.021)	0.010 (0.004)
40	0.483 (0.258)	0.040 (0.004)	0.011 (0.003)
60	1.172 (2.641)	0.047 (0.005)	0.013 (0.002)
80	0.990 (1.138)	0.056 (0.007)	0.014 (0.001)
100	1.074 (1.043)	0.065 (0.007)	0.016 (0.002)
120	1.562 (1.226)	0.073 (0.008)	0.018 (0.003)
140	1.293 (0.615)	0.084 (0.008)	0.020 (0.002)
160	1.802 (1.780)	0.092 (0.009)	0.023 (0.003)
180	2.443 (2.428)	0.098 (0.009)	0.025 (0.002)
200	1.617 (0.773)	0.104 (0.017)	0.026 (0.002)

Table 4
CPU times and key tuning parameters required by PSO and DE to generate various types of locally optimal designs.

Model & criterion	Method	flock size	iteration	cpu time
Compartmental Model <i>D</i> -optimal Design	PSO	20	200	0.4
	DE	20	200	2.9
Compartmental Model <i>c</i> -optimal Design (time to max. concen.)	PSO	60	1000	4.5
	DE	20	200	2.3
Compartmental Model <i>c</i> -optimal Design (AUC)	PSO	40	1000	2.8
	DE	40	200	4.6
Logistic Quadratic Model <i>D</i> -optimal Design	PSO	100	200	1.4
	DE	80	400	57.4
Double Exponential Model <i>D</i> -optimal Design	PSO	40	200	0.9
	DE	20	200	2.9
Survival Model <i>c</i> -optimal Design	PSO	20	100	0.2
	DE	20	100	0.9