Nature-inspired Meta-heuristic Algorithms for Generating Optimal Experimental Designs

Biostatistics 279
Department of Biostatistics, UCLA

Fielding School of Public Health
Spring 2021

UCLA

Outline

1.1 Motivation

1.1 Motivation

- Derivation of optimal designs for nonlinear models is usually tedious, difficult and method for one model does not usually generalize to another

1.1 Motivation

- Derivation of optimal designs for nonlinear models is usually tedious, difficult and method for one model does not usually generalize to another

- Formulae for optimal designs rarely exist and if they do, they are complicated and frequently unhelpful to the practitioners

1.1 Motivation

- Derivation of optimal designs for nonlinear models is usually tedious, difficult and method for one model does not usually generalize to another
- Formulae for optimal designs rarely exist and if they do, they are complicated and frequently unhelpful to the practitioners
- Algorithms are very helpful - available only for some types of optimal designs

1.1 Motivation

- Derivation of optimal designs for nonlinear models is usually tedious, difficult and method for one model does not usually generalize to another
- Formulae for optimal designs rarely exist and if they do, they are complicated and frequently unhelpful to the practitioners
- Algorithms are very helpful - available only for some types of optimal designs
- Proof and speed of convergence, ease of use and availability of software

1.1 Motivation

- Derivation of optimal designs for nonlinear models is usually tedious, difficult and method for one model does not usually generalize to another
- Formulae for optimal designs rarely exist and if they do, they are complicated and frequently unhelpful to the practitioners
- Algorithms are very helpful - available only for some types of optimal designs
- Proof and speed of convergence, ease of use and availability of software
- Is there an easy-to-use and efficient method for finding optimal designs for different types of optimal designs for different types of models?

1.2 General Setup

- a given compact design interval X
- errors are normally and independently distributed
- a parametric model $(f(x), \lambda(x))$ with unknown parameters θ
- a pre-determined sample size N

QUESTION

Given a fixed sample size N, how to select the N points from the design space X to observe the responses y in some optimal way?

1.2 General Setup

- a given compact design interval X
- errors are normally and independently distributed
- a parametric model $(f(x), \lambda(x))$ with unknown parameters θ
- a pre-determined sample size N

<u>QUESTION</u>

Given a fixed sample size N, how to select the N points from the design space X to observe the responses y in some optimal way?

- Focus on continuous or approximate designs (Kiefer, 1958-1982)

1.2 General Setup

- a given compact design interval X
- errors are normally and independently distributed
- a parametric model $(f(x), \lambda(x))$ with unknown parameters θ
- a pre-determined sample size N

QUESTION

Given a fixed sample size N, how to select the N points from the design space X to observe the responses y in some optimal way?

- Focus on continuous or approximate designs (Kiefer, 1958-1982)
- Given N and model, find k (no. of points), $x_1, \ldots, x_k$ (locations) and $w_1, \ldots, w_k$ (proportions) subject to $Nw_1 + \cdots + Nw_k = N$.

1.3 Information Matrices

$$y(x) = f^T(x)\theta + e(x)/\sqrt{\lambda(x)}, \qquad x \in X.$$

- $f(x)$ = a given $d \times 1$ vector of regression functions
- $Ee(x) = 0$; $\quad var(e(x)) = \sigma^2$; $\quad \lambda(x)$ = known positive function

If errors are normally and independently distributed, Fisher information matrix for a k-point design ξ is proportional to

$$M(\xi) = \sum_{i=1}^{k} \lambda(x_i) w_i f(x_i) f^T(x_i), \qquad k \geq d$$

and $\qquad cov(\widehat{\theta}) = M(\xi)^{-1}$(apart from a multiplicative constant).

For a nonlinear model, we have $E(y) = f(x, \theta)$; replace above $f(x)$ by gradient of $f(x, \theta)$ (with respect to θ).

1.4 Optimal Approximate Designs on $X = [-1, 1]$*

Examples of D-optimal designs for estimating model parameters and making inference of the response at an extrapolated dose.

design criterion		linear model		quadratic model		
D-optimality	x_i	-1	1	-1	0	1
	w_i	$1/2$	$1/2$	$1/3$	$1/3$	$1/3$
Extrapolation	x_i	-1	1	-1	0	1
at dose level $z = 2$	w_i	$1/4$	$3/4$	$1/7$	$3/7$	$3/7$

1.4 Optimal Approximate Designs on $X = [-1, 1]^*$

Examples of D-optimal designs for estimating model parameters and making inference of the response at an extrapolated dose.

design criterion		linear model		quadratic model		
D-optimality	x_i	-1	1	-1	0	1
	w_i	$1/2$	$1/2$	$1/3$	$1/3$	$1/3$
Extrapolation	x_i	-1	1	-1	0	1
at dose level $z = 2$	w_i	$1/4$	$3/4$	$1/7$	$3/7$	$3/7$

- *The interval is prototype and scaled appropriately.

1.5 Locally D-optimal Designs for the Logistic Model on $X = [-1, 1]$ (Ford's PhD thesis,1976)

$$log\,\frac{\pi(x)}{1-\pi(x)} = \theta_1 + \theta_2 x, \quad \theta^T = (\theta_1, \theta_2), \quad \theta_1 > 0 \;\;\&\;\; \theta_2 > 0.$$

1.5 Locally D-optimal Designs for the Logistic Model on $X = [-1, 1]$ (Ford's PhD thesis,1976)

$$log \frac{\pi(x)}{1-\pi(x)} = \theta_1 + \theta_2 x, \quad \theta^T = (\theta_1, \theta_2), \quad \theta_1 > 0 \;\; \& \;\; \theta_2 > 0.$$

- Let a solve $exp(z) = (z+1)/(z-1)$, i.e. $a = 1.54$ and let u^* solve

$$exp(\theta_1 + \theta_2 u) = \frac{2 + (u+1)\theta_2}{-2 + (u+1)\theta_2}.$$

1.5 Locally D-optimal Designs for the Logistic Model on $X = [-1, 1]$ (Ford's PhD thesis,1976)

$$log \frac{\pi(x)}{1-\pi(x)} = \theta_1 + \theta_2 x, \quad \theta^T = (\theta_1, \theta_2), \quad \theta_1 > 0 \;\; \& \;\; \theta_2 > 0.$$

- Let a solve $exp(z) = (z+1)/(z-1)$, i.e. $a = 1.54$ and let u^* solve

$$exp(\theta_1 + \theta_2 u) = \frac{2 + (u+1)\theta_2}{-2 + (u+1)\theta_2}.$$

-

condition	locally D-optimal design
$\{\theta : \theta_2 - \theta_1 \geq a\}$	$\{\frac{a-\theta_1}{\theta_2}, \frac{-a-\theta_1}{\theta_2}; \frac{1}{2}, \frac{1}{2}\}$
$\{\theta : \theta_2 - \theta_1 < a, exp(\theta_1 + \theta_2) \leq \frac{\theta_2+1}{\theta_2-1}\}$	$\{-1, u^*; \frac{1}{2}, \frac{1}{2}\}$
$\{\theta : exp(\theta_1 + \theta_2) > \frac{\theta_2+1}{\theta_2-1}\}$	$\{-1, 1; \frac{1}{2}, \frac{1}{2}\}$

1.6 Fedorov's Algorithm for D-optimality (1972)

- Let $k = 0$ and let $\triangle$ be a user-selected precision level, say 0.001.
- Start with a design ξ_k with n support points and $n \geq d$ (dim of f(x)).
- Find information matrix of ξ_k : $M(\xi_k) = \sum_{i=1}^{n} w_i \lambda(x_i) f(x_i) f^T(x_i)$ (*)
- Find $x_k \in X$ that satisfies

$$\lambda(x_k) var(x_k, \xi_k) = max_{x \in X} \lambda(x) var(x, \xi_k) = m_k.$$

- Let $\delta_k = m_k - d$. Stop if $\delta_k < \triangle$ and declare ξ_k is D-optimal (numerically). Otherwise, let ξ^{x_k} be the point mass design at x_k and form a new design ξ_{k+1}:

$$\xi_{k+1} = (1 - p_k)\xi_k + p_k \xi^{x_k}, \qquad p_k = \frac{\delta_k}{(\delta_k + (d-1))d} \in (0, 1).$$

- Set k=k+1, go to (*) and repeat the process until $\delta_k < \triangle$.

1.7 Algorithms: Issues

- Some algorithms can be proved to converge to the optimum

1.7 Algorithms: Issues

- Some algorithms can be proved to converge to the optimum

- Examples are Fedorov's algorithm and related algorithms for finding D, c, L-optimal designs

1.7 Algorithms: Issues

- Some algorithms can be proved to converge to the optimum
- Examples are Fedorov's algorithm and related algorithms for finding D, c, L-optimal designs
- Convergence is usually slow and may not work for more complicated models with more factor or have random effects

1.7 Algorithms: Issues

- Some algorithms can be proved to converge to the optimum
- Examples are Fedorov's algorithm and related algorithms for finding D, c, L-optimal designs
- Convergence is usually slow and may not work for more complicated models with more factor or have random effects
- Periodic collapsing of cluster of points into fewer points is required

1.7 Algorithms: Issues

- Some algorithms can be proved to converge to the optimum
- Examples are Fedorov's algorithm and related algorithms for finding D, c, L-optimal designs
- Convergence is usually slow and may not work for more complicated models with more factor or have random effects
- Periodic collapsing of cluster of points into fewer points is required
- Algorithms may not work for singular optimal designs, designs with fewer than d points

1.7 Algorithms: Issues

- Some algorithms can be proved to converge to the optimum
- Examples are Fedorov's algorithm and related algorithms for finding D, c, L-optimal designs
- Convergence is usually slow and may not work for more complicated models with more factor or have random effects
- Periodic collapsing of cluster of points into fewer points is required
- Algorithms may not work for singular optimal designs, designs with fewer than d points
- Is there an easy-to-use and efficient method for finding optimal designs for different types of optimal designs for any given model?

2 Non-traditional or Modern Algorithms

2 Non-traditional or Modern Algorithms

- Heuristic and meta-heuristic algorithms may offer an answer

- Heuristic and meta-heuristic algorithms may offer an answer
- Widely used in engineering and computer science literature

2 Non-traditional or Modern Algorithms

- Heuristic and meta-heuristic algorithms may offer an answer
- Widely used in engineering and computer science literature
- Statisticians may be most familiar with genetic algorithms, but there are many more out there

2 Non-traditional or Modern Algorithms

- Heuristic and meta-heuristic algorithms may offer an answer
- Widely used in engineering and computer science literature
- Statisticians may be most familiar with genetic algorithms, but there are many more out there
- Nature-inspired meta-heuristic algorithms

2 Non-traditional or Modern Algorithms

- Heuristic and meta-heuristic algorithms may offer an answer
- Widely used in engineering and computer science literature
- Statisticians may be most familiar with genetic algorithms, but there are many more out there
- Nature-inspired meta-heuristic algorithms
- Focus is on Particle Swarm Optimization (PSO) Techniques

2.1 Meta-heuristic Algorithms for Optimal Designs

From Wikipedia, the free encyclopedia: Meta-heuristic

In computer science, meta-heuristic designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Meta-heuristics make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, meta-heuristics do not guarantee an optimal solution is ever found. Many meta-heuristics implement some form of stochastic optimization.

2.1 Meta-heuristic Algorithms for Optimal Designs

From Wikipedia, the free encyclopedia: Meta-heuristic

In computer science, meta-heuristic designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Meta-heuristics make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, meta-heuristics do not guarantee an optimal solution is ever found. Many meta-heuristics implement some form of stochastic optimization.

- Our interest here is nature-inspired meta-heuristic algorithms

2.1 Meta-heuristic Algorithms for Optimal Designs

From Wikipedia, the free encyclopedia: Meta-heuristic

In computer science, meta-heuristic designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Meta-heuristics make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, meta-heuristics do not guarantee an optimal solution is ever found. Many meta-heuristics implement some form of stochastic optimization.

- Our interest here is nature-inspired meta-heuristic algorithms
- Particle Swarm Optimization (PSO) proposed by Eberhard & Kennedy (IEEE, 1995).

Particle swarm optimization: Origins

How can birds or fish exhibit such a coordinated collective behavior?

2.3 Particle Swarm Optimization (PSO)

- Many websites and books provide tutorials, codes and track PSO applications, e.g. http://www.swarmintelligence.org/index.php

2.3 Particle Swarm Optimization (PSO)

- Many websites and books provide tutorials, codes and track PSO applications, e.g. http://www.swarmintelligence.org/index.php
- Lots of application papers in computer science and engineering; more than 14000 hits on http://www.youtube.com

2.3 Particle Swarm Optimization (PSO)

- Many websites and books provide tutorials, codes and track PSO applications, e.g. http://www.swarmintelligence.org/index.php
- Lots of application papers in computer science and engineering; more than 14000 hits on http://www.youtube.com
- Since 2001, at least one annual workshop on Swarm Optimization

2.3 Particle Swarm Optimization (PSO)

- Many websites and books provide tutorials, codes and track PSO applications, e.g. http://www.swarmintelligence.org/index.php
- Lots of application papers in computer science and engineering; more than 14000 hits on http://www.youtube.com
- Since 2001, at least one annual workshop on Swarm Optimization
- International Conference on Swarm Intelligence: Theoretical Advances and Real world Applications in France on June 2011

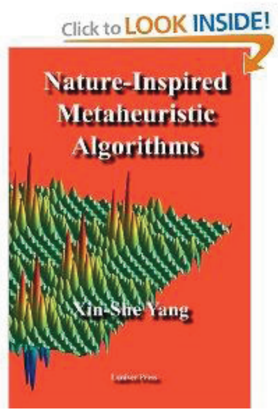
2.3 Particle Swarm Optimization (PSO)

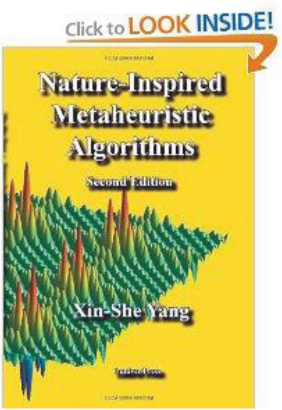
- Many websites and books provide tutorials, codes and track PSO applications, e.g. http://www.swarmintelligence.org/index.php
- Lots of application papers in computer science and engineering; more than 14000 hits on http://www.youtube.com
- Since 2001, at least one annual workshop on Swarm Optimization
- International Conference on Swarm Intelligence: Theoretical Advances and Real world Applications in France on June 2011
- Original book by Eberhard and Kennedy in 2001 on Swarm Intelligence is in its 3^{rd} printing; another book was updated published in 2008 and updated in 2010

2.3 Particle Swarm Optimization (PSO)

- Many websites and books provide tutorials, codes and track PSO applications, e.g. http://www.swarmintelligence.org/index.php
- Lots of application papers in computer science and engineering; more than 14000 hits on http://www.youtube.com
- Since 2001, at least one annual workshop on Swarm Optimization
- International Conference on Swarm Intelligence: Theoretical Advances and Real world Applications in France on June 2011
- Original book by Eberhard and Kennedy in 2001 on Swarm Intelligence is in its 3^{rd} printing; another book was updated published in 2008 and updated in 2010
- A journal, Swarm Intelligence, was born in 2007 and another, International Journal of Swarm Intelligence Research, in 2010

2.4 Xin-She Yang





2.5 Some Applications of PSO

2.5 Some Applications of PSO

- artificial neural network training

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance
- social networks

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance
- social networks
- data mining

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance
- social networks
- data mining
- foraging techniques

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance
- social networks
- data mining
- foraging techniques
- intrusion detection

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance
- social networks
- data mining
- foraging techniques
- intrusion detection
- resources allocation problems

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance
- social networks
- data mining
- foraging techniques
- intrusion detection
- resources allocation problems
- course+exam scheduling in real time

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance
- social networks
- data mining
- foraging techniques
- intrusion detection
- resources allocation problems
- course+exam scheduling in real time
- designing ideotypes for sustainable product systems in genetics

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance
- social networks
- data mining
- foraging techniques
- intrusion detection
- resources allocation problems
- course+exam scheduling in real time
- designing ideotypes for sustainable product systems in genetics
- prediction of stock market indices using hybrid genetic algorithm and PSO with a perturbed term

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance
- social networks
- data mining
- foraging techniques
- intrusion detection
- resources allocation problems
- course+exam scheduling in real time
- designing ideotypes for sustainable product systems in genetics
- prediction of stock market indices using hybrid genetic algorithm and PSO with a perturbed term
- bioinformatics

2.5 Some Applications of PSO

- artificial neural network training
- K-means cluster analysis mathematical finance
- social networks
- data mining
- foraging techniques
- intrusion detection
- resources allocation problems
- course+exam scheduling in real time
- designing ideotypes for sustainable product systems in genetics
- prediction of stock market indices using hybrid genetic algorithm and PSO with a perturbed term
- bioinformatics
- reactive power and voltage control in electric power systems

2.6 Main Features of PSO:

Random generation of an initial population

Each particle has a fitness value that depends on the optimum

Population is reproduced based on fitness value

If requirements are met, stop; otherwise each particle updates its fitness value

Shares similarity with genetic algorithm but differs in important ways discussed in numerous sites such as http://www.alife.org or http://www.engr.iupui.edu/ eberhart with tutorials

2.6 Main Features of PSO:

Random generation of an initial population

Each particle has a fitness value that depends on the optimum

Population is reproduced based on fitness value

If requirements are met, stop; otherwise each particle updates its fitness value

Shares similarity with genetic algorithm but differs in important ways discussed in numerous sites such as http://www.alife.org or http://www.engr.iupui.edu/ eberhart with tutorials

- PSO comprises a very simple concept, its paradigms can be implemented in a few lines of computer code, requires only primitive mathematical operators and is computationally inexpensive in terms of both memory requirement and speed

2.7 Basic Equations and tuning parameters in PSO

$$\mathbf{v}_{i+1} = \omega_i \mathbf{v}_i + c_1\beta_1(\mathbf{p}_i - \mathbf{x}_i) + c_2\beta_2(\mathbf{p}_g - \mathbf{x}_i),$$
$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{v}_i.$$

x_i and v_i: position and velocity for the ith*particle*

β_1 and β_2: random vectors

ω_i: inertia weight that modulates the influence of the former velocity

c_1 and c_2: cognitive learning parameter and social learning parameter

p_i and p_g: Best position for the i^{th} particle (local optimal) and for all particles (global optimal)
For many applications, $c_1 = c_2 = 2$ seem to work well and usually 20 particles suffice (Kennedy, IEEE, 1997).

3 PSO-generated Optimal Designs and Implications

A Locally Optimal Designs for a Compartmental Model

B Locally D-optimal Design for the an Inverse Quadratic Model

C Locally E-optimal Designs and other Minimax Optimal Designs

D Optimal Design for Estimating the Biologically-Optimal Dose for the Continuation Ratio Model

3 PSO-generated Optimal Designs and Implications

A Locally Optimal Designs for a Compartmental Model

- c-optimal Design for Estimating Time to Maximum Concentration

B Locally D-optimal Design for the an Inverse Quadratic Model

C Locally E-optimal Designs and other Minimax Optimal Designs

D Optimal Design for Estimating the Biologically-Optimal Dose for the Continuation Ratio Model

3 PSO-generated Optimal Designs and Implications

A Locally Optimal Designs for a Compartmental Model

- c-optimal Design for Estimating Time to Maximum Concentration
- c-optimal Design for Area Under the Curve

B Locally D-optimal Design for the an Inverse Quadratic Model

C Locally E-optimal Designs and other Minimax Optimal Designs

D Optimal Design for Estimating the Biologically-Optimal Dose for the Continuation Ratio Model

3 PSO-generated Optimal Designs and Implications

A Locally Optimal Designs for a Compartmental Model

- c-optimal Design for Estimating Time to Maximum Concentration
- c-optimal Design for Area Under the Curve
- D-optimal designs for the Compartmental Model

B Locally D-optimal Design for the an Inverse Quadratic Model

C Locally E-optimal Designs and other Minimax Optimal Designs

D Optimal Design for Estimating the Biologically-Optimal Dose for the Continuation Ratio Model

3.1 A 3-parameter Compartment Model

A popular compartmental model with $\theta^T = (\theta_1, \theta_2, \theta_3)$:

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

- Optimality criteria: (i) area under the curve;
  (ii) time to maximum concentration;
  and (iii) maximum concentration.

3.1 A 3-parameter Compartment Model

A popular compartmental model with $\theta^T = (\theta_1, \theta_2, \theta_3)$:

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

- Optimality criteria: (i) area under the curve;
  (ii) time to maximum concentration;
  and (iii) maximum concentration.
- (i) $AUC = \int_0^\infty \eta(t,\theta)dx = \frac{\theta_3}{\theta_2} - \frac{\theta_3}{\theta_1} = g_1(\theta)$

3.1 A 3-parameter Compartment Model

A popular compartmental model with $\theta^T = (\theta_1, \theta_2, \theta_3)$:

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

- Optimality criteria: (i) area under the curve;
  (ii) time to maximum concentration;
  and (iii) maximum concentration.

- (i) $AUC = \int_0^\infty \eta(t,\theta)dx = \frac{\theta_3}{\theta_2} - \frac{\theta_3}{\theta_1} = g_1(\theta)$

- (ii) Time to maximum concentration: $t_{max} = \frac{log\theta_1 - log\theta_2}{\theta_1 - \theta_2} = g_2(\theta)$

3.1 A 3-parameter Compartment Model

A popular compartmental model with $\theta^T = (\theta_1, \theta_2, \theta_3)$:

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

- Optimality criteria: (i) area under the curve;
  (ii) time to maximum concentration;
  and (iii) maximum concentration.

- (i) $AUC = \int_0^\infty \eta(t, \theta) dx = \frac{\theta_3}{\theta_2} - \frac{\theta_3}{\theta_1} = g_1(\theta)$
- (ii) Time to maximum concentration: $t_{max} = \frac{log\theta_1 - log\theta_2}{\theta_1 - \theta_2} = g_2(\theta)$
- (iii) Maximum concentration: $\eta(t_{max}, \theta) = g_3(\theta)$

3.1a Locally c-optimal Designs for a Compartmental Model

These are special cases of L-optimal designs that minimize a given function $g(\theta)$ of the model parameters. Find ξ^* to minimize

$$c^T(\theta)M(\xi,\theta)^{-1}c(\theta)$$

over all designs on the dose interval X, where

$$c(\theta) = \nabla g(\theta) = (\frac{\partial g(\theta)}{\partial \theta_0}, \frac{\partial g(\theta)}{\partial \theta_1}, \frac{\partial g(\theta)}{\partial \theta_2})^T$$

and g is g_1, g_2 or g_3. For this problem, we use nominal values in Atkinson & Donev's (2004) text:

$$\theta_1^0 = 4.29, \qquad \theta_2^0 = 0.0589 \qquad \text{and} \qquad \theta_3^0 = 21.80.$$

3.1a Locally c-optimal Designs for a Compartmental Model

These are special cases of L-optimal designs that minimize a given function $g(\theta)$ of the model parameters. Find ξ^* to minimize

$$c^T(\theta)M(\xi,\theta)^{-1}c(\theta)$$

over all designs on the dose interval X, where

$$c(\theta) = \nabla g(\theta) = (\frac{\partial g(\theta)}{\partial \theta_0}, \frac{\partial g(\theta)}{\partial \theta_1}, \frac{\partial g(\theta)}{\partial \theta_2})^T$$

and g is g_1, g_2 or g_3. For this problem, we use nominal values in Atkinson & Donev's (2004) text:

$$\theta_1^0 = 4.29, \qquad \theta_2^0 = 0.0589 \qquad \text{and} \qquad \theta_3^0 = 21.80.$$

- We will use PSO to find the locally optimal designs!

3.2 Locally D-Optimal Designs for Estimating Inverse Quadratic Models

The model is

$$E(y) = \frac{x+\alpha}{\beta_0 + \beta_1(x+\alpha) + \beta_2(x+\alpha)^2}$$

Examples of the equally weighted locally D-optimal designs:

	nominal values				support points			
Case	α	β_0	β_1	β_2				
(i)	0.1	1.0	-0.8	1	0	0.384	0.964	2.424
(ii)	0.5	1.0	0.8	1	0	0.302	1.285	5.470

Cobby, J. M., Chapman, P. F. and Pike, D. J. (1986). Design of Experiments for Estimating Inverse Quadratic Polynomial Responses, Biometrics, 42, 659 – 664.

Example 3.3 E-optimal designs for the MM model on $X = [0, \tilde{x}]$

The Michaelis-Menten model for a continuous response is

$$y = \frac{\theta_1 x}{\theta_2 + x} + \varepsilon, \quad x > 0 \qquad \theta^T = (\theta_1, \theta_2), \theta_1 > 0, \theta_2 > 0.$$

If ε is normally distributed with mean 0 and constant variance, the Fisher information matrix for a given design ξ is

$$M(\xi, \theta) = \int (\frac{\theta_1 x}{\theta_2 + x})^2 \begin{pmatrix} \frac{1}{\theta_1^2} & -\frac{1}{\theta_1(\theta_2 + x)} \\ -\frac{1}{\theta_1(\theta_2 + x)} & \frac{1}{(\theta_2 + x)^2} \end{pmatrix} d\xi(x).$$

Let

$$w = \frac{\sqrt{2}(\theta_1/\theta_2)^2(1 - \tilde{z})\{\sqrt{2} - (4 - 2\sqrt{2})\tilde{z}\}}{2 + (\theta_1/\theta_2)^2\{\sqrt{2} - (4 - 2\sqrt{2})\tilde{z}\}^2}$$

and $\tilde{z} = \tilde{x}/(\theta_2 + \tilde{x})$. The locally E-optimal design has weight 1-w at $\tilde{x}$ and weight w at $\{(\sqrt{2} - 1)\theta_2\tilde{x}\}/\{2 - \sqrt{2})\tilde{x} + \theta_2\}$ (Dette & Wong, Stat. & Prob. Let., 1999)

Table 2: Locally E-optimal designs for the Michaelis-Menten model on $X = [0, 200]$.

θ_1	θ_2	ξ_{PSO}		E-optimal designs	
100	150	46.52(0.693)	200(0.308)	45.51(0.693)	200(0.307)
100	100	38.15(0.677)	200(0.323)	38.15(0.677)	200(0.323)
100	50	24.78(0.617)	200(0.383)	24.78(0.617)	200(0.383)
100	10	6.52(0.260)	200(0.740)	6.515(0.260)	200(0.740)
100	1	0.70(0.022)	200(0.978)	0.701(0.022)	200(0.978)
10	150	46.50(0.707)	200(0.293)	46.51(0.707)	200(0.293)
10	100	38.14(0.707)	200(0.293)	38.15(0.707)	200(0.293)
10	50	24.78(0.706)	200(0.294)	24.78(0.706)	200(0.294)
10	10	6.52(0.684)	200(0.316)	6.515(0.684)	200(0.316)
10	1	0.70(0.188)	200(0.812)	0.701(0.188)	200(0.812)

Table 2: Locally E-optimal designs for the Michaelis-Menten model on $X = [0, 200]$.

θ_1	θ_2	ξ_{PSO}		E-optimal designs	
100	150	46.52(0.693)	200(0.308)	45.51(0.693)	200(0.307)
100	100	38.15(0.677)	200(0.323)	38.15(0.677)	200(0.323)
100	50	24.78(0.617)	200(0.383)	24.78(0.617)	200(0.383)
100	10	6.52(0.260)	200(0.740)	6.515(0.260)	200(0.740)
100	1	0.70(0.022)	200(0.978)	0.701(0.022)	200(0.978)
10	150	46.50(0.707)	200(0.293)	46.51(0.707)	200(0.293)
10	100	38.14(0.707)	200(0.293)	38.15(0.707)	200(0.293)
10	50	24.78(0.706)	200(0.294)	24.78(0.706)	200(0.294)
10	10	6.52(0.684)	200(0.316)	6.515(0.684)	200(0.316)
10	1	0.70(0.188)	200(0.812)	0.701(0.188)	200(0.812)

- discrepancy stubbornly remained and did not disappear

Table 2: Locally E-optimal designs for the Michaelis-Menten model on $X = [0, 200]$.

θ_1	θ_2	ξ_{PSO}		E-optimal designs	
100	150	46.52(0.693)	200(0.308)	45.51(0.693)	200(0.307)
100	100	38.15(0.677)	200(0.323)	38.15(0.677)	200(0.323)
100	50	24.78(0.617)	200(0.383)	24.78(0.617)	200(0.383)
100	10	6.52(0.260)	200(0.740)	6.515(0.260)	200(0.740)
100	1	0.70(0.022)	200(0.978)	0.701(0.022)	200(0.978)
10	150	46.50(0.707)	200(0.293)	46.51(0.707)	200(0.293)
10	100	38.14(0.707)	200(0.293)	38.15(0.707)	200(0.293)
10	50	24.78(0.706)	200(0.294)	24.78(0.706)	200(0.294)
10	10	6.52(0.684)	200(0.316)	6.515(0.684)	200(0.316)
10	1	0.70(0.188)	200(0.812)	0.701(0.188)	200(0.812)

- discrepancy stubbornly remained and did not disappear
- simply calculation error from the formula; PSO gave right answer!

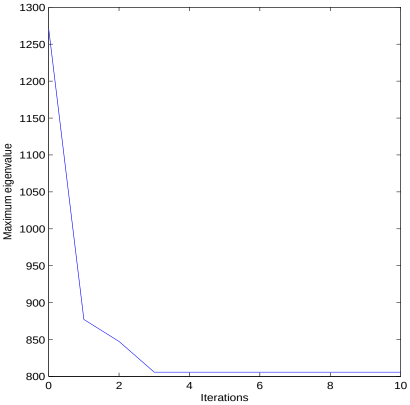


Figure 3: Plot of the maximum eigenvalue of $M(\xi, \theta)^{-1}$ versus the number of PSO iterations in Example 3.

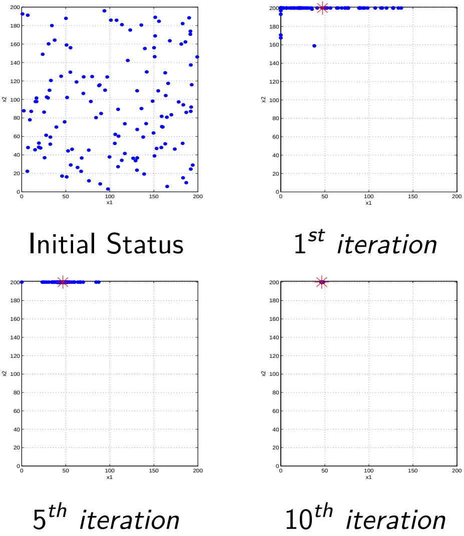
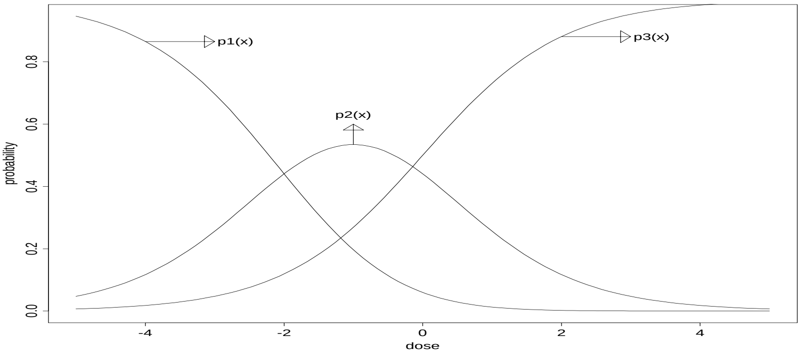


Figure 4: The movement of particles in the PSO search for the E-optimal design for the Michaelis-Menten model at various stages in Example 3. The red star in each of the three plots indicates the current best design.

The Continuation Ratio Model relates probabilities of no response (p_1),efficacy and no severe toxicity (p_2) and severe toxicity (p_3) by:

$$\ln[p_3(\theta,x)/(1-p_3(\theta,x))] = a_1 + b_1 x, \quad b_1 > 0 \qquad (1)$$

$$\ln[p_2(\theta,x)/p_1(\theta,x)] = a_2 + b_2 x, \quad b_2 > 0. \qquad (2)$$



Example 3.5a: Calculus

The biologically optimal dose x_{BOD} depends on $\theta^T = (a_1, b_1, a_2, b_2)$ and solves

$$g(x, \theta) = b_2(1 + e^{-a_1 - b_1 x}) - b_1(1 + e^{a_2 + b_2 x}) = 0.$$

Example 3.5a: Calculus

The biologically optimal dose x_{BOD} depends on $\theta^T = (a_1, b_1, a_2, b_2)$ and solves

$$g(x, \theta) = b_2(1 + e^{-a_1 - b_1 x}) - b_1(1 + e^{a_2 + b_2 x}) = 0.$$

- By the implicit function theorem, the gradient of the solution to the above equation is

$$\left[\frac{\partial g(x_{BOD}(\theta), \theta)}{\partial x}\right]^{-1} \frac{\partial g(x_{BOD}(\theta), \theta)}{\partial \theta}$$
$$= \begin{pmatrix} e^{-a_1 - b_1 x_{BOD}} / [b_1(e^{-a_1 - b_1 x_{BOD}} + e^{a_2 + b_2 x_{BOD}})] \\ x_{BOD} e^{-a_1 - b_1 x_{BOD}} / [b_1(e^{-a_1 - b_1 x_{BOD}} + e^{a_2 + b_2 x_{BOD}})] \\ e^{a_2 + b_2 x_{BOD}} / [b_2(e^{-a_1 - b_1 x_{BOD}} + e^{a_2 + b_2 x_{BOD}})] \\ x_{BOD} e^{a_2 + b_2 x_{BOD}} / [b_2(e^{-a_1 - b_1 x_{BOD}} + e^{a_2 + b_2 x_{BOD}})] \end{pmatrix}.$$

Example 3.5a: Calculus

The biologically optimal dose x_{BOD} depends on $\theta^T = (a_1, b_1, a_2, b_2)$ and solves

$$g(x, \theta) = b_2(1 + e^{-a_1 - b_1 x}) - b_1(1 + e^{a_2 + b_2 x}) = 0.$$

- By the implicit function theorem, the gradient of the solution to the above equation is

$$\left[\frac{\partial g(x_{BOD}(\theta), \theta)}{\partial x}\right]^{-1} \frac{\partial g(x_{BOD}(\theta), \theta)}{\partial \theta}$$
$$= \begin{pmatrix} e^{-a_1 - b_1 x_{BOD}} / [b_1(e^{-a_1 - b_1 x_{BOD}} + e^{a_2 + b_2 x_{BOD}})] \\ x_{BOD} e^{-a_1 - b_1 x_{BOD}} / [b_1(e^{-a_1 - b_1 x_{BOD}} + e^{a_2 + b_2 x_{BOD}})] \\ e^{a_2 + b_2 x_{BOD}} / [b_2(e^{-a_1 - b_1 x_{BOD}} + e^{a_2 + b_2 x_{BOD}})] \\ x_{BOD} e^{a_2 + b_2 x_{BOD}} / [b_2(e^{-a_1 - b_1 x_{BOD}} + e^{a_2 + b_2 x_{BOD}})] \end{pmatrix}.$$

- Use standard algorithm to generate the locally optimal design

3.5b Selected BOD- & D-optimal designs and D-efficiencies (Fan & Chaloner, JSPI, 2003)

dose	weight	(a_1, b_1, a_2, b_2)	dose	weight	D-efficiency
-5.67	0.001	$(-3.3, 0.5, 3.4, 1)$	-4.63	0.292	56%
-0.64	0.800		-1.32	0.416	
4.84	0.199		4.19	0.056	
			8.64	0.236	
-1.26	0.632	$(-1, 0.5, 2, 1)$	-3.54	0.366	67%
4.11	0.368		-0.59	0.403	
			4.80	0.231	
-1.30	0.549	$(-1.04, 0.81.2, 1)$	-2.67	0.370	77%
2.37	0.451		0.00	0.398	
			2.88	0.232	
-14.00	0.100	$(0.4, 0.2, 2, 1)$	-13.00	0.070	62%
-1.14	0.628		-4.11	0.400	
9.99	0.272		-0.77	0.372	
			9.08	0.158	

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)
- Artificial bee colony algorithm (2007)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Gravitational search algorithm (2009)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Gravitational search algorithm (2009)
- Glowworm swarm optimization (2009)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Gravitational search algorithm (2009)
- Glowworm swarm optimization (2009)
- Cuckoo search (Yang & Deb, 2009, Journal of Mathematical Modeling and Numerical Optimization)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Gravitational search algorithm (2009)
- Glowworm swarm optimization (2009)
- Cuckoo search (Yang & Deb, 2009, Journal of Mathematical Modeling and Numerical Optimization)
- Firefly algorithm (2009, 2010)

3.6 Other Nature-Inspired Algorithms

- Ant colony (1991)
- Bees algorithm (2006)
- Invasive weed optimization (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Gravitational search algorithm (2009)
- Glowworm swarm optimization (2009)
- Cuckoo search (Yang & Deb, 2009, Journal of Mathematical Modeling and Numerical Optimization)
- Firefly algorithm (2009, 2010)
- Bat algorithm (2010)

4 Summary

- PSO methodology and its variants, and more generally swarm-based algorithms, are a core of artificial intelligence with great promise to find optimum or a nearly optimum to all kinds of optimization problems. I believe use of such algorithms represents a leap forward in the field of optimal experimental designs.

4 Summary

- PSO methodology and its variants, and more generally swarm-based algorithms, are a core of artificial intelligence with great promise to find optimum or a nearly optimum to all kinds of optimization problems. I believe use of such algorithms represents a leap forward in the field of optimal experimental designs.

- PSO used in bioinformatics but has yet to make an impact in mainstream statistical applications; only 2 talk abstracts appeared in 2011 to find outliers.

4 Summary

- PSO methodology and its variants, and more generally swarm-based algorithms, are a core of artificial intelligence with great promise to find optimum or a nearly optimum to all kinds of optimization problems. I believe use of such algorithms represents a leap forward in the field of optimal experimental designs.

- PSO used in bioinformatics but has yet to make an impact in mainstream statistical applications; only 2 talk abstracts appeared in 2011 to find outliers.

- More realistic optimal designs in this big data era should be more accessible now and hopefully optimal design ideas will be more widely used in practice.

Questions/Comments?

Please send them to Weng Kee Wong

(wkwong@ucla.edu)