Optimal Experimental Designs for Complex or High Dimensional Statistical Models

Weng Kee WONG
Department of Biostatistics, UCLA

Design of Experiments: New Challenges at CIRM
Marseilles, France
April 30-May 4 2018

Outline

1. Motivation and Challenges
2. Nature-inspired Metaheuristic Algorithms
3. Optimal Designs via PSO or its variants
4. Closing Thoughts

Outline

- Motivation and Challenges
- Why Nature-inspired Meta-heuristic Algorithms?
- Sample Applications using PSO or its variants

Outline

- Motivation and Challenges
- Why Nature-inspired Meta-heuristic Algorithms?
- Sample Applications using PSO or its variants
  - Efficient design to estimate Biological Optimal Dose using the Continuation Ratio model (Fan & Chaloner, 2015)

Outline

- Motivation and Challenges
- Why Nature-inspired Meta-heuristic Algorithms?
- Sample Applications using PSO or its variants
  - Efficient design to estimate Biological Optimal Dose using the Continuation Ratio model (Fan & Chaloner, 2015)
  - Minimax and standardized maximin optimal design problems (Chen, Wang, Tung & Wong, Stat. & Comp., 2014)

Outline

- Motivation and Challenges
- Why Nature-inspired Meta-heuristic Algorithms?
- Sample Applications using PSO or its variants
  - Efficient design to estimate Biological Optimal Dose using the Continuation Ratio model (Fan & Chaloner, 2015)
  - Minimax and standardized maximin optimal design problems (Chen, Wang, Tung & Wong, Stat. & Comp., 2014)
  - A Super-Saturated design to maximize $E(s^2)$ (Bulutuglu & Cheng, Annals of Stat., 2004)

Outline

- Motivation and Challenges
- Why Nature-inspired Meta-heuristic Algorithms?
- Sample Applications using PSO or its variants
  - Efficient design to estimate Biological Optimal Dose using the Continuation Ratio model (Fan & Chaloner, 2015)
  - Minimax and standardized maximin optimal design problems (Chen, Wang, Tung & Wong, Stat. & Comp., 2014)
  - A Super-Saturated design to maximize $E(s^2)$ (Bulutuglu & Cheng, Annals of Stat., 2004)
  - Adaptive 2-stage Simon's designs (Kim & Wong, SMMR 2017)

Outline

- Motivation and Challenges
- Why Nature-inspired Meta-heuristic Algorithms?
- Sample Applications using PSO or its variants
  - Efficient design to estimate Biological Optimal Dose using the Continuation Ratio model (Fan & Chaloner, 2015)
  - Minimax and standardized maximin optimal design problems (Chen, Wang, Tung & Wong, Stat. & Comp., 2014)
  - A Super-Saturated design to maximize $E(s^2)$ (Bulutuglu & Cheng, Annals of Stat., 2004)
  - Adaptive 2-stage Simon's designs (Kim & Wong, SMMR 2017)
  - D-optimal designs for high dimensional Logistic regression models (Xu, Wong & Tan, Cybernetika, under review, 2018)

Outline

- Motivation and Challenges
- Why Nature-inspired Meta-heuristic Algorithms?
- Sample Applications using PSO or its variants
  - Efficient design to estimate Biological Optimal Dose using the Continuation Ratio model (Fan & Chaloner, 2015)
  - Minimax and standardized maximin optimal design problems (Chen, Wang, Tung & Wong, Stat. & Comp., 2014)
  - A Super-Saturated design to maximize $E(s^2)$ (Bulutuglu & Cheng, Annals of Stat., 2004)
  - Adaptive 2-stage Simon's designs (Kim & Wong, SMMR 2017)
  - D-optimal designs for high dimensional Logistic regression models (Xu, Wong & Tan, Cybernetika, under review, 2018)
- Closing Thoughts

Outline

1.1 Some personal observations

- Many optimal design problems in the literature for nonlinear models concern 1 or 2 variables;

1.1 Some personal observations

- Many optimal design problems in the literature for nonlinear models concern 1 or 2 variables;
- When there are 3 or more variables, optimal designs are found for nonlinear models when factors are assumed to be addictive;

1.1 Some personal observations

- Many optimal design problems in the literature for nonlinear models concern 1 or 2 variables;
- When there are 3 or more variables, optimal designs are found for nonlinear models when factors are assumed to be addictive;
- High dimensional design problems are very difficult optimization problems to solve analytically or computationally;

1.1 Some personal observations

- Many optimal design problems in the literature for nonlinear models concern 1 or 2 variables;
- When there are 3 or more variables, optimal designs are found for nonlinear models when factors are assumed to be addictive;
- High dimensional design problems are very difficult optimization problems to solve analytically or computationally;
- Dror and Steinberg (Technometrics, 2006) remarked that such design problems with several factors are far from trivial to solve;

1.1 Some personal observations

- Many optimal design problems in the literature for nonlinear models concern 1 or 2 variables;
- When there are 3 or more variables, optimal designs are found for nonlinear models when factors are assumed to be addictive;
- High dimensional design problems are very difficult optimization problems to solve analytically or computationally;
- Dror and Steinberg (Technometrics, 2006) remarked that such design problems with several factors are far from trivial to solve;
- With big data, these are timely problems that offer new challenges.

1.2 Examples

- The theoretical locally D-optimal design for the logistic model with 1 variable was found in Ford's thesis (1972) and reported in the design monograph by Silvey (1980);

1.2 Examples

- The theoretical locally D-optimal design for the logistic model with 1 variable was found in Ford's thesis (1972) and reported in the design monograph by Silvey (1980);
- The results were corrected in Sebastiani & Settimi (JSPI, 1991);

1.2 Examples

- The theoretical locally D-optimal design for the logistic model with 1 variable was found in Ford's thesis (1972) and reported in the design monograph by Silvey (1980);
- The results were corrected in Sebastiani & Settimi (JSPI, 1991);
- An analytical description of the locally D-optimal design for the logistic model with 2 variables and an interaction term was only found in 2017;

1.2 Examples

- The theoretical locally D-optimal design for the logistic model with 1 variable was found in Ford's thesis (1972) and reported in the design monograph by Silvey (1980);
- The results were corrected in Sebastiani & Settimi (JSPI, 1991);
- An analytical description of the locally D-optimal design for the logistic model with 2 variables and an interaction term was only found in 2017;
- Mikaeili (JSPI, 1989) found D-optimal designs for the mixture experiment for the cubic polynomial models in k factors without 3-way effects and 4 years later, extended the results to the case when all 3-way effects on the regular simplex.

1.3 A Minimax Optimal Design Problem

Consider the logistic model on a given design space X:

$$log \frac{\pi(x)}{1-\pi(x)} = \theta_0 + \theta_1 x,$$

where $\theta^T = (\theta_0, \theta_1) \in \Theta$ and contains all plausible values of θ.

Design Criterion: Find $\xi^* = \arg\min_\xi \max_{\theta \in \Theta} log|M(\xi, \theta)|^{-1}$.

1.3 A Minimax Optimal Design Problem

Consider the logistic model on a given design space X:

$$log\frac{\pi(x)}{1-\pi(x)} = \theta_0 + \theta_1 x,$$

where $\theta^T = (\theta_0, \theta_1) \in \Theta$ and contains all plausible values of θ.

Design Criterion: Find $\xi^* = \arg\min_\xi \max_{\theta \in \Theta} log|M(\xi,\theta)|^{-1}$.

- There is no known algorithm that is guaranteed to find a minimax optimal design.

1.3 A Minimax Optimal Design Problem

Consider the logistic model on a given design space X:

$$log \frac{\pi(x)}{1 - \pi(x)} = \theta_0 + \theta_1 x,$$

where $\theta^T = (\theta_0, \theta_1) \in \Theta$ and contains all plausible values of θ.

Design Criterion: Find $\xi^* = \text{arg min}_\xi \ \text{max}_{\theta \in \Theta} \ log|M(\xi, \theta)|^{-1}$.

- There is no known algorithm that is guaranteed to find a minimax optimal design.
- King & Wong (Biometrics, 2002) found minimax D-optimal designs when $\Theta = [0, 3.5] \times [1, 3.5]$ and X is unrestricted:

1.3 A Minimax Optimal Design Problem

Consider the logistic model on a given design space X:

$$log\frac{\pi(x)}{1-\pi(x)} = \theta_0 + \theta_1 x,$$

where $\theta^T = (\theta_0, \theta_1) \in \Theta$ and contains all plausible values of θ.

Design Criterion: Find $\xi^* = \arg\min_\xi \max_{\theta\in\Theta} log|M(\xi,\theta)|^{-1}$.

- There is no known algorithm that is guaranteed to find a minimax optimal design.
- King & Wong (Biometrics, 2002) found minimax D-optimal designs when $\Theta = [0, 3.5] \times [1, 3.5]$ and X is unrestricted:
-

x_i	-0.35	0.62	1.39	2.11	2.88	3.85
w_i	0.18	0.21	0.11	0.11	0.21	0.18

1.4 Sensitivity Plot of the Generated Design

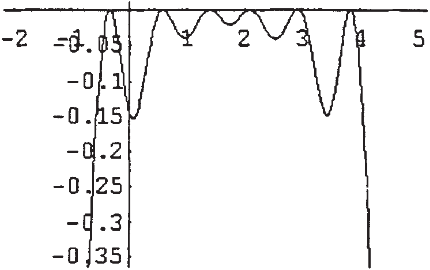


Figure 1. Plot of $\psi(x, \xi^*, \mu^*)$ for example 3.2 with $\Theta = [0, 3.5] \times [1, 3.5]$.

1.4 Sensitivity Plot of the Generated Design

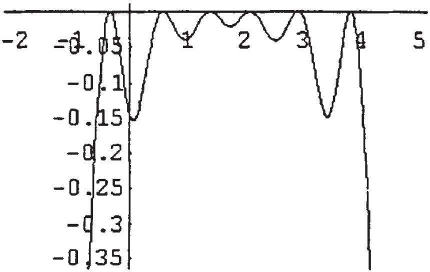


Figure 1. Plot of $\psi(x, \xi^*, \mu^*)$ for example 3.2 with $\Theta = [0, 3.5] \times [1, 3.5]$.

- References: Wong (Biometrika, 1992), Wong & Cook (JRSSB, 1993), Berger, King & Wong (Psychometrika, 1994)

1.4 Sensitivity Plot of the Generated Design

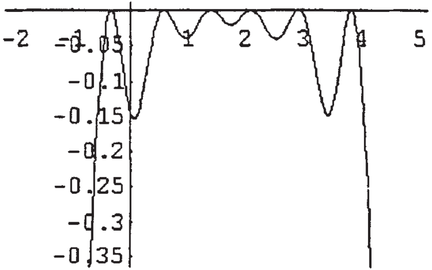


Figure 1. Plot of $\psi(x, \xi^*, \mu^*)$ for example 3.2 with $\Theta = [0, 3.5] \times [1, 3.5]$.

- References: Wong (Biometrika, 1992), Wong & Cook (JRSSB, 1993), Berger, King & Wong (Psychometrika, 1994)
- What about standardized maximin designs?

1.5 Approaches to solving design problems

1.5 Approaches to solving design problems

- An analytical approach can be limiting, for example, results stated for the design interval $[-1, 1]$ (and does not discuss what happens for other intervals);

1.5 Approaches to solving design problems

- An analytical approach can be limiting, for example, results stated for the design interval $[-1, 1]$ (and does not discuss what happens for other intervals);
- Proof invariably breaks down if the model is slightly changed;

1.5 Approaches to solving design problems

- An analytical approach can be limiting, for example, results stated for the design interval $[-1, 1]$ (and does not discuss what happens for other intervals);
- Proof invariably breaks down if the model is slightly changed;
- Proof is valid usually under a few technical conditions;

1.5 Approaches to solving design problems

- An analytical approach can be limiting, for example, results stated for the design interval $[-1, 1]$ (and does not discuss what happens for other intervals);
- Proof invariably breaks down if the model is slightly changed;
- Proof is valid usually under a few technical conditions;
- Proof is just too daunting;

1.5 Approaches to solving design problems

- An analytical approach can be limiting, for example, results stated for the design interval $[-1, 1]$ (and does not discuss what happens for other intervals);
- Proof invariably breaks down if the model is slightly changed;
- Proof is valid usually under a few technical conditions;
- Proof is just too daunting;
- When criterion is convex, can keep on guessing and check optimality - is a loser's game!;
- For example, find a design on the design space to maximize

$$\Psi(\xi) = \min_{\theta \in \Theta} \left\{ \frac{|M(\xi, \theta)|}{\sup_{\gamma} |M(\gamma, \theta)|} \right\}^{1/p}.$$

1.5 Approaches to solving design problems

- An analytical approach can be limiting, for example, results stated for the design interval $[-1, 1]$ (and does not discuss what happens for other intervals);
- Proof invariably breaks down if the model is slightly changed;
- Proof is valid usually under a few technical conditions;
- Proof is just too daunting;
- When criterion is convex, can keep on guessing and check optimality - is a loser's game!;
- For example, find a design on the design space to maximize

$$\Psi(\xi) = \min_{\theta \in \Theta} \left\{ \frac{|M(\xi, \theta)|}{\sup_{\gamma} |M(\gamma, \theta)|} \right\}^{1/p}.$$

- A practical way is to develop effective and flexible algorithms;

1.5 Approaches to solving design problems

- An analytical approach can be limiting, for example, results stated for the design interval $[-1, 1]$ (and does not discuss what happens for other intervals);
- Proof invariably breaks down if the model is slightly changed;
- Proof is valid usually under a few technical conditions;
- Proof is just too daunting;
- When criterion is convex, can keep on guessing and check optimality - is a loser's game!;
- For example, find a design on the design space to maximize

$$\Psi(\xi) = \min_{\theta \in \Theta} \left\{ \frac{|M(\xi, \theta)|}{\sup_\gamma |M(\gamma, \theta)|} \right\}^{1/p}.$$

- A practical way is to develop effective and flexible algorithms;
- Many algorithms have their issues; work on $[-1, 1]$, but may not $[-1, 10]$, let alone $[-1, 1]^k$ or $[2, 10]^k$

1.6 Time required to discretize a 10-dimensional search space with different number of equally spaced points using a Mac laptop 2.6 GHz Intel Core i5

number of equally spaced points per covariate space	total number of grid points	CPU time required to generate the grid (secs)
2	$2^{10} = 1024$	0.0067
3	$3^{10} = 59049$	0.2302
4	$4^{10} = 1,048,576$	3.1136
5	$5^{10} = 9,765,625$	27.5529
6	$6^{10} = 60,466,176$	172.2832
7	$7^{10} = 282,475,249$	848.2922

1.6 Time required to discretize a 10-dimensional search space with different number of equally spaced points using a Mac laptop 2.6 GHz Intel Core i5

number of equally spaced points per covariate space	total number of grid points	CPU time required to generate the grid (secs)
2	$2^{10} = 1024$	0.0067
3	$3^{10} = 59049$	0.2302
4	$4^{10} = 1,048,576$	3.1136
5	$5^{10} = 9,765,625$	27.5529
6	$6^{10} = 60,466,176$	172.2832
7	$7^{10} = 282,475,249$	848.2922

- There are immediate implications....

1.7 Traditional algorithms for high dimensional problems?

- Current design algorithms such as Cocktail-based algorithms: Yu (Stat. & Comp. 2011) and Yang, et al. (JASA, 2014) may not work WELL for high dimensional problems.

1.7 Traditional algorithms for high dimensional problems?

- Current design algorithms such as Cocktail-based algorithms: Yu (Stat. & Comp. 2011) and Yang, et al. (JASA, 2014) may not work WELL for high dimensional problems.
- Mathematical programming tools that require the search space be discretized or solvers like semi-definite programming (SDP) (Papp, JASA, 2011, Duarte & Wong, Stat. & Comp., 2014, Duarte, Wong & Atkinson, J. of Multivariate Analysis, 2015 and Duarte, Wong & Dette, Stat. & Comp., 2017, Duarte, Sagnol & Wong, 2018, CSDA) may become inapplicable.

1.8 Mathematistry

In Praise of Simplicity not Mathematistry! Ten Simple Powerful Ideas for the Statistical Scientist

Roderick J. LITTLE

Ronald Fisher was by all accounts a first-rate mathematician, but he saw himself as a scientist, not a mathematician, and he railed against what George Box called (in his Fisher lecture) "mathematistry." Mathematics is the indispensable foundation of statistics, but for me the real excitement and value of our subject lies in its application to other disciplines. We should not view statistics as another branch of mathematics and favor mathematical complexity over clarifying, formulating, and solving real-world problems. Valuing simplicity, I describe 10 simple and powerful ideas that have influenced my thinking about statistics, in my areas of research interest: missing data, causal inference, survey sampling, and statistical modeling in general. The overarching theme is that statistics is a missing data problem and the goal is to predict unknowns with appropriate measures of uncertainty.

KEY WORDS: Calibrated Bayes; Causal inference; Measurement error; Missing data; Penalized spline of propensity.

1. INTRODUCTION: THE UNEASY RELATIONSHIP BETWEEN STATISTICS AND MATHEMATICS

American Statistical Association President, Sastry Pantula, recently proposed renaming the Division of Mathematical Sciences at the U.S. National Science Foundation as the Division of Mathematical and Statistical Sciences. Opponents, who viewed statistics as a branch of mathematics, questioned why statistics should be singled out for special treatment.

Data can be assembled in support of the argument that statistics *is* different—for example, the substantial number of academic departments of statistics and biostatistics, the rise of the statistics advanced placement examination, and the substantial number of undergraduate statistics majors. But the most important factor for me is that statistics is not just a branch of mathematics. It is an inductive method, defined by its applications to the sciences and other areas of human endeavor, where we try to glean information from data.

The relationship between mathematics and statistics is somewhat uneasy. Since the mathematics of statistics is often viewed as basically rather pedestrian, statistics is rather low on the totem pole of mathematical subdisciplines. Statistics needs its mathematical parent, since it is the indispensable underpinning of the subject. On the other hand, unruly statistics has ambitions to reach beyond the mathematics fold: it comes alive in applica- and medicine, and with increasing influence recently on the hard sciences such as astronomy, geology and physics.

The scientific theme of modern statistics fits the character of its most influential developer, the great geneticist, R. A. Fisher, who seemed to revolutionize the field of statistics in his spare time! Fisher's momentous move to Rothampsted Experimental Station rather than academia underlined his dedication to science. Though an excellent mathematician, Fisher viewed himself primarily as a scientist, and disparaged rivals like Neyman and Pearson as mere "mathematicians."

George Box's engaging Fisher lecture focused on the links between statistics and science (Box 1976). He wrote:

> My theme then will be first to show the part that [Fisher] being a good scientist played in his astonishing ingenuity, originality, inventiveness, and productivity as a statistician, and second to consider what message that has for us now.

Box attributed Fisher's hostility to mathematicians to distaste for what he called "mathematistry," which he defined as

> [. . .] the development of theory for theory's sake, which, since it seldom touches down with practice, has a tendency to redefine the problem rather than solve it. Typically, there has once been a statistical problem with scientific relevance but this has long since been lost sight of. (Box 1976)

Mathl. Comput. Modelling Vol. 18, No. 11, pp. 29–57, 1993
Printed in Great Britain.

0895-7177/93 $6.00 + 0.00

Simulated Annealing: Practice versus Theory

L. INGBER
Lester Ingber Research
P.O.B. 857, McLean, VA 22101, U.S.A.
ingber@alumni.caltech.edu

(Received June 1993; accepted July 1993)

Abstract—Simulated annealing (SA) presents an optimization technique with several striking positive and negative features. Perhaps its most salient feature, statistically promising to deliver an optimal solution, in current practice is often spurned to use instead modified faster algorithms, "simulated quenching" (SQ). Using the author's Adaptive Simulated Annealing (ASA) code, some examples are given which demonstrate how SQ can be much faster than SA without sacrificing accuracy.

Keywords—Simulated annealing, Random algorithm, Optimization technique.

1. INTRODUCTION

1.1. Shades of Simulated Annealing

Simulated annealing presents an optimization technique that can:

(a) process cost functions possessing quite arbitrary degrees of nonlinearities, discontinuities, and stochasticity;
(b) process quite arbitrary boundary conditions and constraints imposed on these cost functions;
(c) be implemented quite easily with the degree of coding quite minimal relative to other nonlinear optimization algorithms;
(d) statistically guarantee finding an optimal solution.

Section 2 gives a short introduction to SA, emphasizing its property of (weak) ergodicity. Note that for very large systems, ergodicity is not an entirely rigorous concept when faced with the real task of its computation [1]. Moreover, in this paper "ergodic" is used in a very weak sense, as it is not proposed, theoretically or practically, that all states of the system are actually to be visited.

Even "standard" SA is not without its critics. Some negative features of SA are that it can:

(A) be quite time-consuming to find an optimal fit, especially when using the "standard" Boltzmann technique;
(B) be difficult to fine tune to specific problems, relative to some other fitting techniques;
(C) suffer from "over-hype" and faddish misuse, leading to misinterpretation of results; and

Many of the authors cited here generously responded to my electronic mail requests for (p)reprints on current work in this field; quite a few read earlier drafts and contributed their feedback. Their timely response and helpful suggestions are gratefully acknowledged. Graphs were produced using XVGR (graphics for exploratory data analysis), a public domain software package running under UNIX and X11, developed by Paul Turner at the Oregon Graduate Institute.

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

Outline

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free
- easy to program and implement; in fact many codes are widely and freely available

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free
- easy to program and implement; in fact many codes are widely and freely available
- powerful and fast, and SIMPLE

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free
- easy to program and implement; in fact many codes are widely and freely available
- powerful and fast, and SIMPLE
- produce the optimal design neatly

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free
- easy to program and implement; in fact many codes are widely and freely available
- powerful and fast, and SIMPLE
- produce the optimal design neatly
- does not need to discretize the search space

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free
- easy to program and implement; in fact many codes are widely and freely available
- powerful and fast, and SIMPLE
- produce the optimal design neatly
- does not need to discretize the search space
- can solve high dimensional optimization problems

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free
- easy to program and implement; in fact many codes are widely and freely available
- powerful and fast, and SIMPLE
- produce the optimal design neatly
- does not need to discretize the search space
- can solve high dimensional optimization problems
- optimal designs with various number of support points

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free
- easy to program and implement; in fact many codes are widely and freely available
- powerful and fast, and SIMPLE
- produce the optimal design neatly
- does not need to discretize the search space
- can solve high dimensional optimization problems
- optimal designs with various number of support points
- can find singular designs all the same

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free
- easy to program and implement; in fact many codes are widely and freely available
- powerful and fast, and SIMPLE
- produce the optimal design neatly
- does not need to discretize the search space
- can solve high dimensional optimization problems
- optimal designs with various number of support points
- can find singular designs all the same
- can find optimal design when assumed conditions are not met

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free
- easy to program and implement; in fact many codes are widely and freely available
- powerful and fast, and SIMPLE
- produce the optimal design neatly
- does not need to discretize the search space
- can solve high dimensional optimization problems
- optimal designs with various number of support points
- can find singular designs all the same
- can find optimal design when assumed conditions are not met
- can help find analytic solution or formula of optimal design for complicated problems

2.1 Why Nature-Inspired Metaheuristic Algorithms?

- flexible, applicable to a very broad class of optimization problems
- assumptions free
- easy to program and implement; in fact many codes are widely and freely available
- powerful and fast, and SIMPLE
- produce the optimal design neatly
- does not need to discretize the search space
- can solve high dimensional optimization problems
- optimal designs with various number of support points
- can find singular designs all the same
- can find optimal design when assumed conditions are not met
- can help find analytic solution or formula of optimal design for complicated problems
- projection based versus population based approach for

2.2 Usage of Nature-Inspired Metaheuristic Algorithms

- Recent trends indicate rapid growth of nature-inspired optimization in academia and industry. (Whitacre, 2011, Computing, Vol. 93, 121-133.)

- Survival of the flexible: explaining the recent dominance of nature-inspired optimization within a rapidly evolving world. (Whitacre, 2011, Computing, Vol. 93, 135-146.)

2.2 Usage of Nature-Inspired Metaheuristic Algorithms

- Recent trends indicate rapid growth of nature-inspired optimization in academia and industry. (Whitacre, 2011, Computing, Vol. 93, 121-133.)
- Survival of the flexible: explaining the recent dominance of nature-inspired optimization within a rapidly evolving world. (Whitacre, 2011, Computing, Vol. 93, 135-146.)
- Nature-inspired metaheuristic algorithms can lead in the new frontier of research and solve optimization problems with millions of variables (Foreword by editors in a special issue in Information Sciences, 2015, Vol. 316, 437-439.)

2.3 Metaheuristic Algorithms

From Wikipedia, the free encyclopedia: Metaheuristic

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

2.3 Metaheuristic Algorithms

From Wikipedia, the free encyclopedia: Metaheuristic

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

- Our interest here is nature-inspired metaheuristic algorithms

2.3 Metaheuristic Algorithms

From Wikipedia, the free encyclopedia: Metaheuristic

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

- Our interest here is nature-inspired metaheuristic algorithms
- Particle Swarm Optimization (PSO) method is based on animal instincts (Eberhard & Kennedy, IEEE, 1995)

2.4 PSO (Kennedy & Eberhard, 1995)

2.4 PSO (Kennedy & Eberhard, 1995)

- PSO is a component of Swarm intelligence as part of artificial intelligence in Biologically Inspired Engineering. The field studies collective behavior in biological systems and how such behaviors can be applied to computing and robotics.

2.5 Particle Swarm Optimization (PSO)

- Lots of applications in computer science and engineering; more than 14000 hits on http://www.youtube.com

2.5 Particle Swarm Optimization (PSO)

- Lots of applications in computer science and engineering; more than 14000 hits on http://www.youtube.com

- Since 2001, one or more annual workshops on Swarm Optimization

2.5 Particle Swarm Optimization (PSO)

- Lots of applications in computer science and engineering; more than 14000 hits on http://www.youtube.com
- Since 2001, one or more annual workshops on Swarm Optimization
- International Conference on Swarm Intelligence: Theoretical Advances and Real world Applications in France on June 2011

2.5 Particle Swarm Optimization (PSO)

- Lots of applications in computer science and engineering; more than 14000 hits on http://www.youtube.com
- Since 2001, one or more annual workshops 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

2.5 Particle Swarm Optimization (PSO)

- Lots of applications in computer science and engineering; more than 14000 hits on http://www.youtube.com
- Since 2001, one or more annual workshops 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
- A journal, Swarm Intelligence, was born in 2007 and another, International Journal of Swarm Intelligence Research, in 2010

2.6 Basic Equations and Tuning Parameters in PSO

Two defining equations:

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

x_i and v_i: position and velocity for the i^{th} particle
β_1 and β_2: random vectors
ω_i: inertia weight that modulates the influence of the last velocity
c_1: cognitive learning parameter
c_2: social learning parameter
p_i: Best position for the i^{th} particle (local optimal)
p_g : Best position for all particles (global optimal)

For many applications, $c_1 = c_2 = 2$ seem to work well and usually $20 - 50$ particles will suffice (Kennedy, IEEE, 1997).

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Glowworm swarm optimization (2009)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Glowworm swarm optimization (2009)
- Cuckoo search (2009)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Glowworm swarm optimization (2009)
- Cuckoo search (2009)
- Firefly algorithm (2009, 2010)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Glowworm swarm optimization (2009)
- Cuckoo search (2009)
- Firefly algorithm (2009, 2010)
- Bat algorithm (2010)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Glowworm swarm optimization (2009)
- Cuckoo search (2009)
- Firefly algorithm (2009, 2010)
- Bat algorithm (2010)
- Grey Wolf algorithm (2014,2016)

2.7 Other Nature-Inspired Meta-Heuristic Algorithms

- Ant colony (1991)
- Differential Evolutionary (1997)
- Bees algorithm (2006)
- Artificial bee colony algorithm (2007)
- Saplings growing-up algorithm (2007)
- Monkey search (2007)
- Viral Systems (2008)
- Intelligent water drops algorithm (2009)
- Glowworm swarm optimization (2009)
- Cuckoo search (2009)
- Firefly algorithm (2009, 2010)
- Bat algorithm (2010)
- Grey Wolf algorithm (2014,2016)
- Competitive Swarm Optimizer for large scale computing (2014)

2.8 Examples of Variants of Particle Swarm Optimization

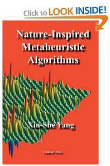
- Hierarchical PSO (Applications of Evol. Comput., 2004)
- Quantum PSO (Evolutionary Computation, 2004)
- Unified PSO (Advances in Natural Computation, 2005)
- Tournament PSO (IEEE Symposium Proceedings, 2007)
- Ladder PSO (Applied Soft Computing, 2009)
- Globally Convergent Particle Swarm Optimization via Branch-and-Bound (Computer and Info Science, 2010)
- Strength Pareto PSO (Evolutionary Computation, 2010)
- Set-Based PSO (IEEE Transactions on Evol. Comp., 2010)
- Catfish PSO (Artificial Intelligence Research, 2012)
- Compact PSO (Information Sciences, 2013)
- Human Behavior-based PSO (Scientific World Journal, 2014)
- Selectively Informed PSO (Scientific Reports, 2014)
- Competitive Swarm Optimizer (Cybernetika, 2014)
- Fast PSO (Soft Computing, 2015)
- Galactic Swarm Optimization (Applied Soft Computing, 2016)

2.9 Resources for Metaheuristic Optimization

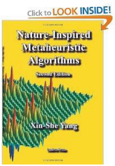
Scholarpedia, the peer-reviewed open-access encyclopedia:
http://www.scholarpedia.org/article/Metaheuristic_Optimization

Many PSO tutorials and information at
http://www.swarmintelligence.org/index.php or
https://www.youtube.com/watch?v=sB1n9a9yxJk

Xin-She Yang's 2008 book and updated in 2010:



2008



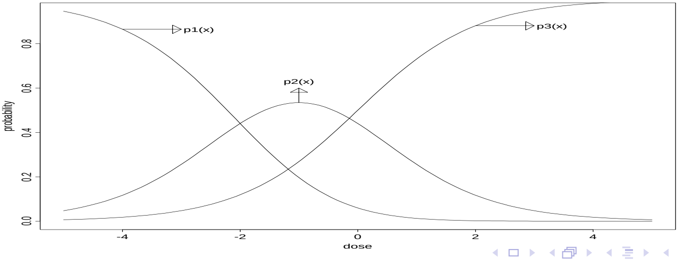
2010 (2nd edition)

Outline

3.1 Design to estimate the Biological Optimal Dose (BOD)

Fan & Chaloner (JSPI, 2001) used the Continuation Ratio Model to relate probabilities of no response (p_1), efficacy and no severe toxicity (p_2) and severe toxicity (p_3) by

$$\begin{aligned} \log[p_3(\theta,x)/(1-p_3(\theta,x))] &= a_1 + b_1 x, \quad b_1 > 0 \\ \log[p_2(\theta,x)/p_1(\theta,x)] &= a_2 + b_2 x, \quad b_2 > 0. \end{aligned} \tag{1}$$



3.2 Calculus - inverse/implicit function theorem

If $\theta^T = (a_1, b_1, a_2, b_2)$, the BOD x_{BOD} solves

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

3.2 Calculus - inverse/implicit function theorem

If $\theta^T = (a_1, b_1, a_2, b_2)$, the BOD x_{BOD} 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, desired vector d in the c-optimality criterion $d^T M(\xi)^{-1} d$ is

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

3.2 Calculus - inverse/implicit function theorem

If $\theta^T = (a_1, b_1, a_2, b_2)$, the BOD x_{BOD} 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, desired vector d in the c-optimality criterion $d^T M(\xi)^{-1} d$ is

$$\begin{aligned} d &= \left[\frac{\partial g(x_{max}(\theta),\theta)}{\partial x}\right]^{-1} \frac{\partial g(x_{max}(\theta),\theta)}{\partial \theta} \\ &= \begin{pmatrix} e^{-a_1-b_1x_{max}}/[b_1(e^{-a_1-b_1x_{max}} + e^{a_2+b_2x_{max}})] \\ x_{max}e^{-a_1-b_1x_{max}}/[b_1(e^{-a_1-b_1x_{max}} + e^{a_2+b_2x_{max}})] \\ e^{a_2+b_2x_{max}}/[b_2(e^{-a_1-b_1x_{max}} + e^{a_2+b_2x_{max}})] \\ x_{max}e^{a_2+b_2x_{max}}/[b_2(e^{-a_1-b_1x_{max}} + e^{a_2+b_2x_{max}})] \end{pmatrix}. \end{aligned}$$

- PSO generates the locally D and c-optimal designs

3.3 PSO for multiple-objective optimal designs

If both D-optimal and c-optimal designs are sought but each has unequal interest, PSO can find a multiple-objective optimal design. Ideas from Cook and Wong (JASA,1994):

- Prioritize the importance of the criteria and formulate them as convex functionals to be minimized
- Rewrite the problem as a constrained optimal design problem
- Impose a specified minimum efficiency required for the more important objective:
- Solve the compound optimal design problem obtained by taking a convex combination of all the criteria
- For each set of convex weights, the problem is now single objective and PSO finds the optimum easily
- An efficiency plot can then identify the sought after constrained optimal design. Qiu, Chen, Wang & Wong (Swarm and Evolutionary Computation Journal, 2014).

3.4 Standardized Maximin Optimal Designs

- The maximin approach assumes a known plausible region Θ for the model parameters θ. The maximin optimal design maximizes the smallest determinant of the $p \times p$ information matrix among all $\theta \in \Theta$.

- The standardized maximin D-optimal design maximizes

$$\Psi(\xi) = \min_{\theta \in \Theta} \left\{ \frac{|M(\xi, \theta)|}{\sup_{\gamma} |M(\gamma, \theta)|} \right\}^{1/p},$$

where $M(\gamma, \theta)$ is the $p \times p$ Information matrix for the nonlinear model with parameter θ from design γ.

3.5 Standardized Maximin Optimal Designs

Chen, Chang, Wang & Wong (Statistics & Computing, 2014) found minimax or maximin optimal designs for several types of nonlinear models when the denominator is ignored.

Bogacka et al. (JBS, 2011) considered 4 inhibition enzyme-kinetic models and found locally D-optimal designs. Chen, Chen, Chen & Wong (Chemo. Intell. Lab. Sys., 2017) used PSO to (i) find standardized maximin optimal designs, (ii) showed they are not always minimally supported, and (iii) from the equivalence theorem, determined formulae of the standardized maximin optimal designs for the 3-parameter inhibition kinetic models.

3.6 $E(s^2)$-optimal Super-Saturated Design (SSD)

Booth and Cox (Technometrics, 1962) proposed the $E(s^2)$-criterion to minimize the average nonorthogonality between all pairs of columns in the design matrix X, i.e.

$$E(s^2) = \sum_{i<j} s_{ij}^2 / \binom{p}{2},$$

where s_{ij} is the dot product between the ith and jth columns of X. A theoretical lower bound for the design criterion for a SSD with p-factors and N-runs is available:

$$E(s^2) \geq \frac{p - N + 1}{(p-1)(N-1)} N^2$$

.

A Swarm Intelligence based method was used to find $E(s^2)$-optimal SSDs for much higher values of N and p. (Phoa, Chen, Wang & Wong, Technometrics, 2014.)

3.7 Extended 2-stage Adaptive Designs

In Simon 2-Stage design for Phase II trials, user first selects two efficacy rates of interest p_0 and p_1 with $p_0 < p_1$.

- Set up hypothesis: $H_O : p \leq p_0$ versus $H_1 : p > p_1$

3.7 Extended 2-stage Adaptive Designs

In Simon 2-Stage design for Phase II trials, user first selects two efficacy rates of interest p_0 and p_1 with $p_0 < p_1$.

- Set up hypothesis: $H_O : p \leq p_0$ versus $H_1 : p > p_1$
- Determine 4 positive integers subject to type 1 and type 2 error constraints:
    - number of patients in Stage 1

3.7 Extended 2-stage Adaptive Designs

In Simon 2-Stage design for Phase II trials, user first selects two efficacy rates of interest p_0 and p_1 with $p_0 < p_1$.

- Set up hypothesis: $H_O : p \leq p_0$ versus $H_1 : p > p_1$
- Determine 4 positive integers subject to type 1 and type 2 error constraints:
  - number of patients in Stage 1
  - number of responders in Stage 1

3.7 Extended 2-stage Adaptive Designs

In Simon 2-Stage design for Phase II trials, user first selects two efficacy rates of interest p_0 and p_1 with $p_0 < p_1$.

- Set up hypothesis: $H_O : p \leq p_0$ versus $H_1 : p > p_1$
- Determine 4 positive integers subject to type 1 and type 2 error constraints:
  - number of patients in Stage 1
  - number of responders in Stage 1
  - number of (additional) patients in Stage 2

3.7 Extended 2-stage Adaptive Designs

In Simon 2-Stage design for Phase II trials, user first selects two efficacy rates of interest p_0 and p_1 with $p_0 < p_1$.

- Set up hypothesis: $H_O : p \leq p_0$ versus $H_1 : p > p_1$
- Determine 4 positive integers subject to type 1 and type 2 error constraints:
  - number of patients in Stage 1
  - number of responders in Stage 1
  - number of (additional) patients in Stage 2
  - number of (additional) responders in Stage 2

3.7 Extended 2-stage Adaptive Designs

In Simon 2-Stage design for Phase II trials, user first selects two efficacy rates of interest p_0 and p_1 with $p_0 < p_1$.

- Set up hypothesis: $H_O : p \leq p_0$ versus $H_1 : p > p_1$
- Determine 4 positive integers subject to type 1 and type 2 error constraints:
    - number of patients in Stage 1
    - number of responders in Stage 1
    - number of (additional) patients in Stage 2
    - number of (additional) responders in Stage 2
- Apply a greedy search to solve the discrete optimization problem relating Binomial probabilities and error rates

3.7 Extended 2-stage Adaptive Designs

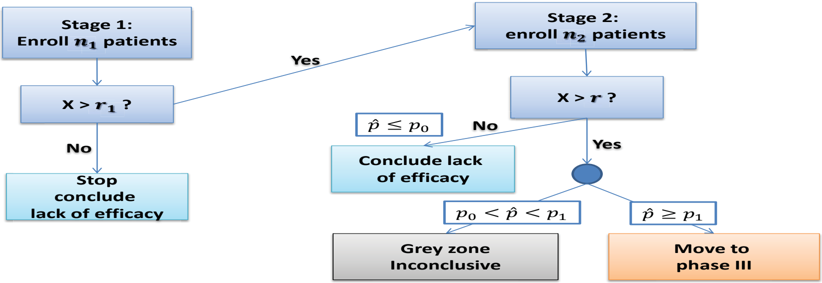
In Simon 2-Stage design for Phase II trials, user first selects two efficacy rates of interest p_0 and p_1 with $p_0 < p_1$.

- Set up hypothesis: $H_O : p \leq p_0$ versus $H_1 : p > p_1$
- Determine 4 positive integers subject to type 1 and type 2 error constraints:
  - number of patients in Stage 1
  - number of responders in Stage 1
  - number of (additional) patients in Stage 2
  - number of (additional) responders in Stage 2
- Apply a greedy search to solve the discrete optimization problem relating Binomial probabilities and error rates
- Lin & Shih (Biometrics, 2004) generalized the problem to 2 alternative hypotheses, and we extended it to 3 sets of alternative hypotheses.

3.8 A Discrete Optimization Problem

Simon's Two-Stage Designs

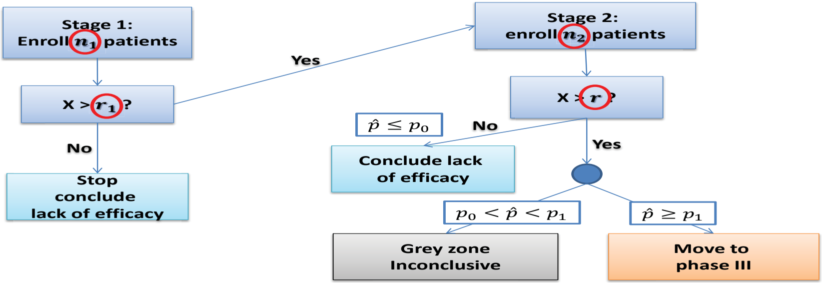
- X: the number of responders



3.8 A Discrete Optimization Problem (cont'd)

Simon's Two-Stage Designs

- X: the number of responders



3.9 Test capability of PSO

Simon's 2-stage design has 4 parameters and the criterion was t o minimize the expected sample size, or minimize the maximum sample size for the whole trial.

Goal: Extend Simon's 2 stage designs for 3 alternatives target efficacy rates

3.9 Test capability of PSO

Simon's 2-stage design has 4 parameters and the criterion was t o minimize the expected sample size, or minimize the maximum sample size for the whole trial.

Goal: Extend Simon's 2 stage designs for 3 alternatives target efficacy rates

- Kim & Wong (SMMR, 2018) applied a modified version of PSO and searched over a constrained 10-dimensional space of positive integers and found optimal designs that a greedy algorithm cannot.

3.10 A 10 Integer-valued Parameters Problem to Optimize

Problem is to optimize $\theta^T = (s_1, r_1, q_1, n_1, s, l, r, m, q, n)$ given error for testing each of the three possible alternative hypotheses rates and the criterion is one of minimizing the maximum (or expected) sample sizes.

The parameters l, m, n are the total number of patients required for the entire trial corresponding to the alternative hypotheses, H_{11}: $p > p_1$, H_{12}: $p > p_2$, and H_{13}: $p > p_3$, respectively.

If true response probability is p, similar argument in Simon's original paper shows the probability of failing to reject H_0 is

$$G(\theta|p) = B(s_1, n_1, p) + \sum_{x=s_1+1}^{\min(r_1,s)} b(x, n_1, p)B(s-x, l_2, p) + \sum_{x=r_1+1}^{\min(q_1,r)} b(x, n_1, p)B(r-x, m_2, p) + \sum_{x=q_1+1}^{\min(q,n_1)} b(x, n_1, p)B(q-x, n_2, p),$$

3.11 Three-Objective Optimal Designs for the Hill's Model

Assume nominal values, dose interval and the minimum effect sought δ are given for the Hill's model. For a user-selected vector $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ with $\lambda_i \geq 0, i = 1, 2, 3$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$, the sought multiple-objective optimal design is the approximate design that maximizes

$$\lambda_1 log(Eff_D(\xi)) + \lambda_2 log(Eff_{ED_{50}}(\xi)) + \lambda_3 log(Eff_{MED}(\xi))$$

$$= \lambda_1 0.25 log(|M(\xi, \boldsymbol{\Theta})|) - \lambda_2 log(Var(\widehat{ED}_{50})) - \lambda_3 log(Var(\widehat{MED})).$$

Here $ED50$ and MED are the median effective dose and the user-specified minimum effective dose.

3.11 Three-Objective Optimal Designs for the Hill's Model

Assume nominal values, dose interval and the minimum effect sought δ are given for the Hill's model. For a user-selected vector $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ with $\lambda_i \geq 0, i = 1, 2, 3$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$, the sought multiple-objective optimal design is the approximate design that maximizes

$$\lambda_1 log(Eff_D(\xi)) + \lambda_2 log(Eff_{ED_{50}}(\xi)) + \lambda_3 log(Eff_{MED}(\xi))$$

$$= \lambda_1 0.25 log(|M(\xi, \boldsymbol{\Theta})|) - \lambda_2 log(Var(\widehat{ED}_{50})) - \lambda_3 log(Var(\widehat{MED})).$$

Here *ED*50 and *MED* are the median effective dose and the user-specified minimum effective dose.

- Reference: Hyun, S. W., Wong, W. K. and Yang, Y. (2018). VNM: A R Package for Finding Multiple-Objective Optimal Designs for the 4-Parameter Logistic Model. Journal of Statistical Software. In press.

3.12 Sample Applications of PSO to Find Optimal Designs

- Qiu, J. H., Chen, R. B., Wang, W.C. & Wong, W. K. (2014). Using Animal Instincts to Design Efficient Biomedical Studies. Swarm and Evolutionary Computation Journal.
- Chen, R. B., Wang, W.C., Chang, & Wong, W. K. (2014). Minimax Optimal Designs via Particle Swarm Optimization Methods. Statistics & Computing.
- Wong, W. K., Wang, W.C., Chang, C. & Chen, R. B. (2015). Optimal Designs for Mixture Models using Particle Swarm Optimization Methods. PlosOne.
- Phoa, K. H. F., Chen, R. B., Wang, W. C. & Wong, W. K. (2015). Optimizing Two-level Supersaturated Designs by Particle Swarm. Technometrics.
- Chen, R. B., Chen P. Y., Tung, H. C. and Wong, W. K. (2015). Exact D-optimal Designs for Michaelis-Menten Model with Correlated Observations by Particle Swarm Optimization.

3.12 Applications of PSO to Find Optimal Designs (cont'd)

- Chen, P. Y., Chen, R. B., Tung, H. C. and Wong, W. K. (2017). Standardized Maximim *D*-optimal Designs for Enzyme Kinetic Inhibition Models. Chemometrics and Intelligent Laboratory System.
- Kim, S. and Wong, W. K. (2017). Extended Two-stage Adaptive Designs for Phase II Clinical Trials. Statistical Methods in Medical Research.
- Masoudi, E., Holling, H. and Wong, W. K. (2017). Application of **Imperialist Competitive Algorithm** to Find Minimax Optimal Designs. Computational Statistics and Data Analysis.
- Lukemire, J., Mandal, A. and Wong, W. K. (2018). *d*-QPSO: A **Quantum-Behaved Particle Swarm** Technique for Finding *D*-Optimal Designs for Models with Mixed Factors and a Binary Response. Technometrics, in press.

3.13 Other applications in statistical journals

Paterlini, S. and Krink, T.
Differential evolution and particle swarm optimisation in partitional clustering
Computational and Statistical Data Analysis, 2006

Leathermann, E. R., Dean, A. M. and Santer, T. J.
Designing combined physical and computer experiments to maximize prediction accuracy
Computational and Statistical Data Analysis, 2017

Chen, R. B, Hsieh, D. N., Hung, Y. and Wang, W.
Optimizing Latin hypercube designs by particle swarm
Stat & Computing, 2016

Mak, S. and Roshan, J.
Minimax and Minimax projection designs using clustering
Journal of Computation and Graphic Statistics, 2018

3.14 Locally D-optimal design on $[-1,1]^4$ for a logistic model having factors x_1, x_2, x_3, x_4 and all pairwise interactions. Nominal values for the parameters are $\theta_0 = 2.15, \theta_1 = -0.76, \theta_2 = 0.33, \theta_3 = 2.73, \theta_4 = 1.42, \theta_5 = 1.89, \theta_6 = -2.39, \theta_7 = 2.57, \theta_8 = 0.65, \theta_9 = 0.58, \theta_{10} = -2.45$.

x_1	x_2	x_3	x_4	w
-1.000	-1.000	-1.000	-1.000	0.064
-1.000	-1.000	-0.307	-1.000	0.042
-1.000	-1.000	1.000	1.000	0.094
-1.000	1.000	-1.000	-0.963	0.087
-1.000	1.000	-0.066	1.000	0.083
-0.984	1.000	1.000	-1.000	0.086
-0.169	-1.000	-1.000	1.000	0.093
0.908	1.000	-1.000	1.000	0.094
1.000	-1.000	-0.640	-1.000	0.088
1.000	-1.000	1.000	0.567	0.089
1.000	0.472	-1.000	-1.000	0.089
1.000	1.000	1.000	1.000	0.091

3.15 A locally *D*-optimal design found by Competitive Swarm Optimizer (CSO) for a five-factor Poisson model with all pairwise interaction terms (Zhang and Wong, under review)

x_1	x_2	x_3	x_4	x_5	w_i
1.000	1.000	0.685	1.000	-0.730	0.033
1.000	1.000	0.430	-1.000	-1.000	0.062
1.000	-1.000	-1.000	1.000	1.000	0.010
1.000	0.011	1.000	1.000	-1.000	0.062
1.000	1.000	1.000	-0.460	-1.000	0.063
1.000	1.000	0.677	1.000	-1.000	0.057
0.404	1.000	0.670	1.000	-1.000	0.058
1.000	-0.470	1.000	1.000	-0.581	0.061
0.406	1.000	1.000	-0.454	-1.000	0.063
1.000	-0.479	0.508	1.000	-1.000	0.062
-1.000	-1.000	-1.000	1.000	1.000	0.048
1.000	1.000	1.000	1.000	-0.724	0.056
1.000	0.405	1.000	0.127	-1.000	0.062
0.390	-0.012	1.000	1.000	-1.000	0.062
1.000	1.000	1.000	-0.601	-0.691	0.063
1.000	1.000	1.000	1.000	-1.000	0.061
0.337	1.000	1.000	1.000	-0.686	0.057
0.411	1.000	1.000	1.000	-1.000	0.059

With small weights, a large sample is required to implement the design.

3.16 Optimal Discrimination Designs for 2 or 3 multi-factor polynomial models without a known null model assumption (Yue, Vanderburgh & Wong, under review)

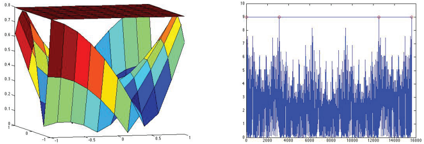


Figure 1: Plots of the sensitivity functions of two designs found by our algorithm for Example 3 (left) and Example 5 (right) to confirm their optimality.

Outline

4.1 Summary

- Remember the "No Free Lunch Theorem" (Wolpert & Macready, IEEE Trans. on Evolutionary Comput., 1997)

4.1 Summary

- Remember the "No Free Lunch Theorem" (Wolpert & Macready, IEEE Trans. on Evolutionary Comput., 1997)
- PSO is not limited to minimizing convex functionals, can find exact designs,minimum bias designs, minimum mean-square error (MSE) designs (Lukemire, Mandal & Wong, Technometrics, 2018, Stokes, Mandal & Wong, 2018, under prep.)

4.1 Summary

- Remember the "No Free Lunch Theorem" (Wolpert & Macready, IEEE Trans. on Evolutionary Comput., 1997)
- PSO is not limited to minimizing convex functionals, can find exact designs,minimum bias designs, minimum mean-square error (MSE) designs (Lukemire, Mandal & Wong, Technometrics, 2018, Stokes, Mandal & Wong, 2018, under prep.)
- Hybridize metaheuristic algorithms with mathematical programming methods (such as simplex methods,Interior Point, etc) to speed up the search for the optimum (Garcia-Rodenas, Fidalgo-Lopez & Wong, 2018, under review)

4.1 Summary

- Remember the "No Free Lunch Theorem" (Wolpert & Macready, IEEE Trans. on Evolutionary Comput., 1997)
- PSO is not limited to minimizing convex functionals, can find exact designs,minimum bias designs, minimum mean-square error (MSE) designs (Lukemire, Mandal & Wong, Technometrics, 2018, Stokes, Mandal & Wong, 2018, under prep.)
- Hybridize metaheuristic algorithms with mathematical programming methods (such as simplex methods,Interior Point, etc) to speed up the search for the optimum (Garcia-Rodenas, Fidalgo-Lopez & Wong, 2018, under review)
- Find MLEs and identify parameter redundancy in a complex statistician distribution (Park & Wong, 2018, under prep.)

4.2 PSO for Solving a System of Nonlinear Equations

Let $x^T = (x_1, x_2, .., x_n)$. We want to solve

$$f_1(x_1, x_2, .., x_n) = 0,$$
$$f_2(x_1, x_2, .., x_n) = 0,$$
$$\cdot$$
$$\cdot$$

and

$$f_r(x_1, x_2, .., x_n) = 0.$$

4.2 PSO for Solving a System of Nonlinear Equations

Let $x^T = (x_1, x_2, .., x_n)$. We want to solve

$$f_1(x_1, x_2, .., x_n) = 0,$$
$$f_2(x_1, x_2, .., x_n) = 0,$$
$$.$$
$$.$$

and

$$f_r(x_1, x_2, .., x_n) = 0.$$

- If x^* is the global minimum of $F(x) = \sum_{i=1}^{r} f_i^{\ 2}(x)$, then x^* solves the above system of equations. (Jabeipour, et al., 2011, Computers and Mathematics with Applications)

4.2 PSO for Solving a System of Nonlinear Equations

Let $x^T = (x_1, x_2, .., x_n)$. We want to solve

$$f_1(x_1, x_2, .., x_n) = 0,$$
$$f_2(x_1, x_2, .., x_n) = 0,$$
$$\cdot$$
$$\cdot$$

and

$$f_r(x_1, x_2, .., x_n) = 0.$$

- If x^* is the global minimum of $F(x) = \sum_{i=1}^{r} f_i^{\,2}(x)$, then x^* solves the above system of equations. (Jabeipour, et al., 2011, Computers and Mathematics with Applications)
- Alternatively, assume $r = n$ and define $F(x) = \sum_{i=1}^{r} |f_i(x)|$ and find its global minimum. (Wang, et al., 2009, IEEE Xplore)

4.3 Current work

- Optimal discrimination design problems (Chen, Chen, Chen & Wong, Technometrics, under revision)

4.3 Current work

- Optimal discrimination design problems (Chen, Chen, Chen & Wong, Technometrics, under revision)
- Optimal designs for PKPD models with random effects (Thanks to France Mentre's team)

4.3 Current work

- Optimal discrimination design problems (Chen, Chen, Chen & Wong, Technometrics, under revision)
- Optimal designs for PKPD models with random effects (Thanks to France Mentre's team)
- Applying DE to generate generalized minimum aberration designs (Thanks to Boxin and Xu's guidance)

4.3 Current work

- Optimal discrimination design problems (Chen, Chen, Chen & Wong, Technometrics, under revision)
- Optimal designs for PKPD models with random effects (Thanks to France Mentre's team)
- Applying DE to generate generalized minimum aberration designs (Thanks to Boxin and Xu's guidance)
- Design for a nuclear reactor using the Langmuir-Hinshelwood model (Chen, Contescu, Mee & Wong, under prep.)

4.4 Langmuir-Hinshelwood model

Mechanism of graphite oxidation by water in its various form using the rate equation of the oxidation reaction

$$r = \frac{k_1 P_{H_20}}{1 + k_2(P_{H_2})^2 + k_3 P_{H_20}} = f(H_20, H_2, T, \theta),$$

where $\theta^T = (A_1, A_2, A_3, E_1, E_2, E_3)$ is the vector of model parameter, $k_i = A_i Exp(-E_i/RT), i = 1, 2, 3$ and$R = 8.314$ is a universal gas constant. The design variables are temperature (T), water pressure (PH_2) and water vapor (P_{H_2O}).

Goal: Find an extrapolation optimal design for inference at low range values of the design variables for the model

$$logr = logf((H_20, H_2, T, \theta) + \epsilon,$$

and ϵ $N(0, \sigma^2)$ and all errors are independent.

4.5 Conclusions

- There are many more meta-heuristic algorithms, for example, Imperialist Competitive Algorithm (ICA) (Masoudi, Holling & Wong, CSDA, 2016, Journal of Computation and Graphical Statistics, 2018, to appear)

4.5 Conclusions

- There are many more meta-heuristic algorithms, for example, Imperialist Competitive Algorithm (ICA) (Masoudi, Holling & Wong, CSDA, 2016, Journal of Computation and Graphical Statistics, 2018, to appear)

- **Acknowledgements**: The research of Wong was partially supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award Number R01GM107639

4.5 Conclusions

- There are many more meta-heuristic algorithms, for example, Imperialist Competitive Algorithm (ICA) (Masoudi, Holling & Wong, CSDA, 2016, Journal of Computation and Graphical Statistics, 2018, to appear)
- **Acknowledgements**: The research of Wong was partially supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award Number R01GM107639
- Thank you; please send comments to wkwong@ucla.edu

4.6 Acknowledgments of Collaborators

Ray-Bing Chen, Ping Yang Chen (Dept. of Statistics, National Cheng Kung University, Taiwan)
Belmiro Duarte (Dept. of Chemical Engineering, Coimbra University, Portugal)
Ricardo Garcia-Ródenas(Dept. of Mathematics, UCLM, Spain)
Won Hyun (Johnson & Johnson, Irvine, California)
Jesus Fidalgo-Lopez (Dept. of Narva University, Spain)
Joshua Lukemire (Dept. of Biostatistics, Emory University)
Abhyudal Mandal (Dept. of Statistics, U of Georgia)
Ehsan Masoudi, Heinz Holling (Dept. of Psychology, Münster University, Germany)
Guanghao Qi (Dept. of Biostatistics, Johns Hopkins University)
Yu Shi, Zizhao Zhao (Dept. of Biostatistics, UCLA)
Weichung Wang (Dept. of Mathematics, National Taiwan University)