

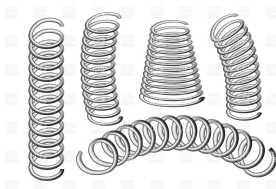
Discrete optimal design and construction algorithms

(These are commonly used tools for designing physical experiments with parametric models.)

ISyE 8813 - Lecture slides

Motivation

Designed experiments vs. observational data



- Consider first a heat treatment process on truck leaf springs (Pignatiello and Ramberg, 1985)
- Four **experimental factors**, each with two levels:
 - Furnace temperature
 - Heating time
 - Transfer time
 - Hold-down time
- Goal is to estimate the **response surface** of the spring's free height using 16 experimental runs (budget constraint)
- Should these runs be **randomly allocated** or **picked judiciously**?

Designed experiments vs. observational data



- Consider next a study on the effect of certain exposure factors (smoking, diet, etc.) on a **rare** strain of brain cancer, with an occurrence rate less than 0.01%
- Should the data for such a study be obtained by:
 - **Observing** records from a cancer treatment center?
 - **Subjecting** individuals to different exposure factors, and seeing whether they have cancer after a few years?

Designed experiments vs. observational data



Advantages of designed experiments:

- Optimal performance in terms of parameter inference and prediction accuracy
- Uniform coverage of the design space, thereby reducing the need for extrapolation

Disadvantages of designed experiments:

- Undesirable when the cost of generating data far outweighs the cost of observing data
- The process of generating data may violate certain ethical principles in the social sciences

Optimal design criteria

Linear model

For illustration, consider the simple **linear model** for data:

$$y_i = \sum_{j=1}^p x_{ij} \beta_j + \epsilon_i, \quad i = 1, \dots, n,$$

where:

- $\mathbf{y} = (y_1, \dots, y_n) \in \mathbb{R}^n$ is the vector of observed **responses**,
- $\mathbf{x}_j = (x_{1j}, \dots, x_{nj}) \in \mathbb{R}^n$ is the j -th **covariate** vector,
- $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_p) \in \mathbb{R}^{n \times p}$ is the **model matrix**,
- $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p) \in \mathbb{R}^p$ is the **coefficient** vector,
- $\{\epsilon_i\}_{i=1}^n \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma^2)$ is the **observation noise**.

Linear model

- Consider the **least-squares estimator**:

$$\hat{\beta}_{LS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Check the following as an exercise:

- (Unbiasedness):**

$$\mathbb{E}[\hat{\beta}_{LS}] = \beta$$

- (Variance):**

$$V[\hat{\beta}_{LS}] = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$$

- By the **Gauss-Markov** theorem, the estimator variance $V[\hat{\beta}_{LS}]$ is **minimal** amongst all unbiased linear estimators of β .
- Goal is to carefully **design** \mathbf{X} so that $V[\hat{\beta}_{LS}]$ is **minimized** over all possible model matrices

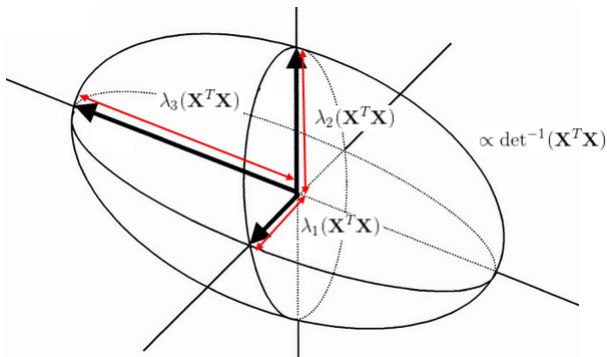
A closer look at $\mathbf{X}^T \mathbf{X}$

- The $100(1 - \alpha)\%$ **confidence region** of β is:

$$R_\alpha = \{\xi : \sigma^{-2}(\xi - \hat{\beta}_{LS})^T (\mathbf{X}^T \mathbf{X})(\xi - \hat{\beta}_{LS}) \leq c_\alpha\},$$

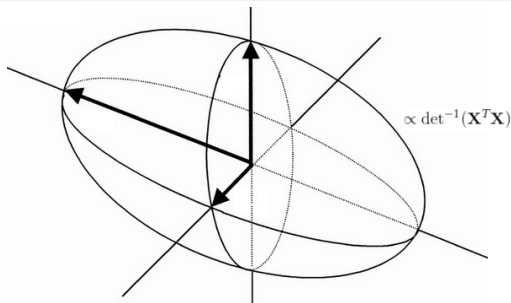
where c_α is some cut-off depending on α

- R_α is an **ellipsoid** whose shape depends on $\mathbf{X}^T \mathbf{X}$:



- This **geometry** provides insight on several popular designs

D-optimality: maximizing volume

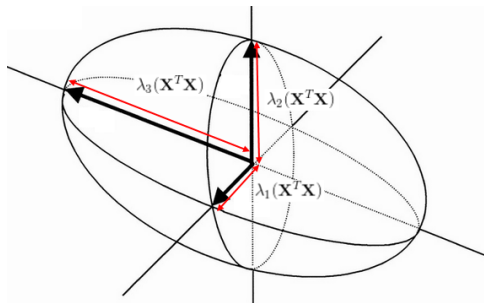


- A **D-optimal** design is defined as:

$$\operatorname{argmax}_{\mathbf{X}} \det^{-1}(\mathbf{X}^T \mathbf{X}) = \operatorname{argmin}_{\mathbf{X}} \det(\mathbf{X}^T \mathbf{X})$$

- **Geometric interpretation:** Volume of ellipsoid is proportional to $\det^{-1}(\mathbf{X}^T \mathbf{X})$
- **Statistical interpretation:** D-optimal designs maximize the confidence region volume for β

A-optimality: maximizing principle axes lengths



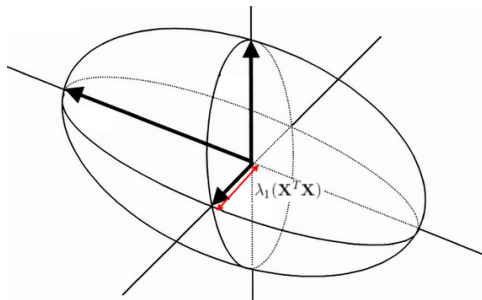
- An **A-optimal** design is defined as:

$$\operatorname{argmin}_{\mathbf{X}} \operatorname{tr} \{ (\mathbf{X}^T \mathbf{X})^{-1} \} = \operatorname{argmin}_{\mathbf{X}} \sum_{j=1}^p \lambda_j \{ (\mathbf{X}^T \mathbf{X})^{-1} \},$$

where λ_j is the j -th smallest eigenvalue

- **Geometric interpretation:** Principal axes of ellipsoid have lengths $\lambda_j(\mathbf{X}^T \mathbf{X})$
- **Statistical interpretation:** A-optimal designs minimize the sum of marginal variances for β

E-optimality: maximizing smallest eigenvalue



- An **E-optimal** design is defined as:

$$\operatorname{argmax}_{\mathbf{X}} \lambda_1(\mathbf{X}^T \mathbf{X})$$

- **Geometric interpretation:** $\lambda_1(\mathbf{X}^T \mathbf{X})$ is the length of the **shortest** principle axis of ellipsoid
- **Statistical interpretation:** E-optimal designs minimize the **contrast with maximum variance** for β

G- and I-optimality: minimizing predictive error



- Under the linear model, the **fitted values** at training inputs are:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{LS}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

- A **G-optimal design** is defined as:

$$\underset{\mathbf{X}}{\operatorname{argmin}} \max [\operatorname{diag}\{\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\}]$$

- Minimizes the **maximum prediction variance** over design space
- An **I-optimal design** aims to minimize the **average prediction variance** over design space

Construction algorithms

Construction algorithms



- Since these designs are defined as the **minimizer** of a criterion, its construction algorithms typically involve some **optimization**
- For discrete design spaces, standard integer-programming methods can be **computationally expensive**
- Statisticians and applied mathematicians have developed **efficient construction heuristics** which work well in practice.

These include:

- Exchange algorithms
- Simulated annealing
- Genetic algorithms
- Particle swarm optimization

Exchange algorithms



- First proposed in Federov (1972) and Wynn (1972), **exchange algorithms (EAs)** are efficient heuristics for generating large designs with many factors
- Starting from an initial design, EAs exchange a design point (or design coordinate) with a new point (or coordinate) only if such an exchange results in an improvement in the design criterion

Exchange algorithms

Visualization of point-exchange and coordinate-exchange algorithms:

| Run | Factor | | | | | | |
|-----|--------|---|---|---|---|---|-----|
| | A | B | C | D | E | F | new |
| 1 | + | + | - | + | + | + | - |
| 2 | + | - | + | + | + | - | - |
| 3 | - | + | + | + | - | - | - |
| 4 | + | + | + | - | - | - | + |
| 5 | + | + | - | - | - | + | - |
| 6 | + | - | - | - | + | - | + |
| 7 | - | - | - | + | - | + | + |
| 8 | - | - | + | - | + | + | - |
| 9 | - | + | - | + | + | - | + |
| 10 | + | - | + | + | - | + | + |
| new | - | - | - | - | - | - | - |

Iteratively exchange design rows (points) or columns (coordinates) to improve objective criterion

Exchange algorithms

Popular **point-exchange** algorithms:

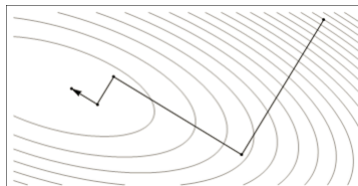
- **Federov algorithm** (Federov, 1972) for D-optimal designs:
 - When exchanging a current point \mathbf{x}_{old} with \mathbf{x}_{new} , the **multiplicative change** in $\det(\mathbf{X}^T \mathbf{X})$ is $1 + \Delta(\mathbf{x}_{\text{old}}, \mathbf{x}_{\text{new}})$, where:

$$\Delta(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}', \mathbf{x}') - [d(\mathbf{x}, \mathbf{x})d(\mathbf{x}', \mathbf{x}') - d(\mathbf{x}, \mathbf{x}')^2] - d(\mathbf{x}, \mathbf{x}),$$

$$\text{and } d(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}$$

- The Federov algorithm **searches** through the possible exchanges for all n points, then **exchanges** the point which provides the largest reduction in $\det(\mathbf{X}^T \mathbf{X})$
 - A similar procedure holds for A-optimal designs
- **Improvements** on the Federov algorithm includes the modified Federov algorithm (Cook and Nachtsheim, 1980) and the k-exchange algorithm (Johnson and Nachtsheim, 1983)

Exchange algorithms



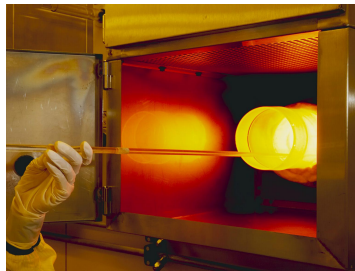
Popular **coordinate-exchange** algorithms:

- Meyer and Nachtsheim (1995):
 - Coordinate EAs avoid the enumeration of large candidate sets, and provides considerable **reductions in computation time**
 - A similar **multiplicative change** $1 + \tilde{\Delta}(\mathbf{c}_{\text{old}}, \mathbf{c}_{\text{new}})$ is observed when exchanging coordinate \mathbf{c}_{old} with \mathbf{c}_{new}
 - Coordinate EAs are **widely used in SAS/JMP software**

EAs have a close connection with **coordinate descent algorithms** for **convex** programs (see Tseng, 2001)

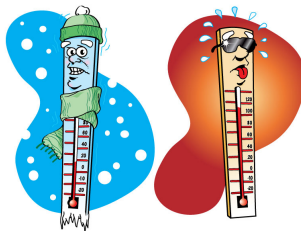
- Exercise: D-, A-, E-, G- and I-optimality criteria are convex

Simulated annealing



- **Simulated annealing (SA)** is a stochastic algorithm for **global optimization**, first proposed by Khachaturyan et al. (1979)
- Motivated by the **metallurgy annealing** process - the **heating** and **cooling** of material to increase crystal size and reduce defects
- An adaptation of the popular **Metropolis-Hastings algorithm** (Metropolis et al., 1953) for sampling complex distributions

Simulated annealing



SA follows the steps below:

- Initialize a high **temperature** $T > 0$ and initial solution x_0
- Starting from solution x with objective $f(x)$, randomly **propose** a new solution x' with some probability $P(x'|x)$
- Sample $z \sim \mathcal{U}[0, 1]$, and **accept** the proposal x' if:

$$z < \exp\{(f(x) - f(x'))/T\}$$

- Gradually **decrease** T to focus on low-energy solutions, i.e., with low objective values

Simulated annealing

Visualization of SA for one-dimensional function:

Simulated annealing

For **difficult** optimization problems, SA often **outperforms** the usual gradient descent algorithms. It is therefore well-suited for design optimization:

- Haines (1987), Meyer and Nachtsheim (1988) proposed **generalized** SA algorithms for constructing exact D-optimal designs
- Lejeune (2003) developed a heuristic **combining SA and exchange** algorithms to efficiently compute approximate D-optimal designs
- Angelis et al. (2011) introduced a modified SA variant for generating optimal exact D- and A-optimal designs with **correlated** errors

Particle swarm optimization

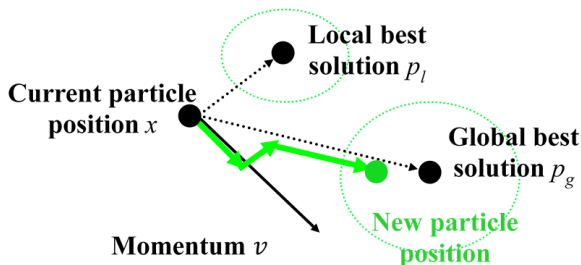


- **Particle swarm optimization (PSO)** is a derivative-free optimization algorithm proposed by Eberhart and Kennedy (1995)
- PSO mimics the behavior of **bird flocks** in locating a food source: each bird searches for food along its own flight path, but also in regions explored by the flock

Particle swarm optimization

PSO starts with a “swarm” of solutions particles, then iteratively guides each particle towards:

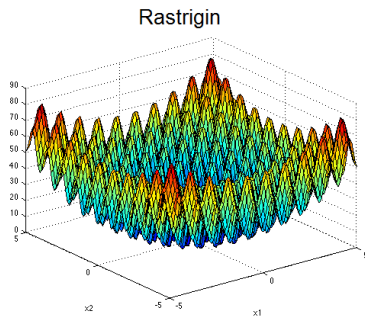
- **Global-best particle**: Best solution over swarm
- **Local-best particle**: Best solution along its own path
- **Momentum** from previous position



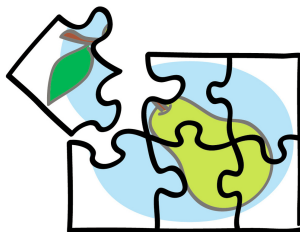
Particle swarm optimization

Visualization of PSO on the Rastrigin function:

$$f(\mathbf{x}) = 10d + \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i)]$$

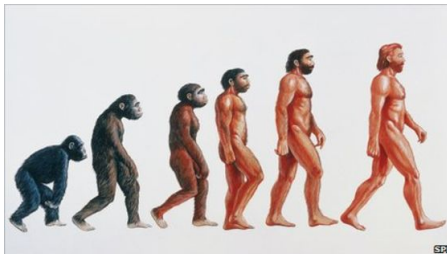


Particle swarm optimization



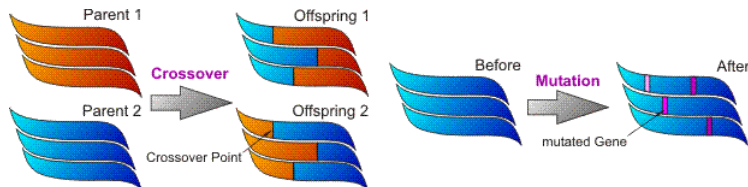
- One appeal of PSO is that it can easily be **combined** (or **hybridized**) with other design optimization methods
- This hybridization has been a topic of recent interest:
 - Wang et al. (2012): Combined PSO with **simplex projections** to find A- and D-optimal mixture designs
 - Lukemire et al. (2016+): Employed PSO with **equivalence theorem** checks to construct D-optimal mixed-factor designs
 - Mak and Joseph (2017): Proposed a modified **PSO-clustering** algorithm for generating minimax and minimax projection designs

Genetic algorithms



- **Genetic algorithms (GAs)** are **optimization heuristics** belonging to a larger class of evolutionary algorithms, and have been employed since the pioneering work of John Holland in the 1970's
- Inspiration comes from the principle of **natural selection**: genes from fitter individuals are **recombined** and **mutated** to form future generations, while weaker ones are **eliminated**

Genetic algorithms



Beginning with a “pool” of solutions (typically coded as **binary strings**), GAs perform three key operations:

- **Selection** (survival of the fittest): At each iteration, **remove** solutions with poor objective values
- **Crossover** (mating): New solutions are generated from **swapping** two solutions up to a randomly-chosen bit
- **Mutation** (random modifications): With some low probability, new solutions from crossovers will have some bits **flipped**

Genetic algorithms



The **flexibility** of GAs allows it to be a useful tool (in conjunction with other methods) for design construction:

- Safadi and Wang (1991) employed GAs for constructing mixed multi-level orthogonal arrays
- Broudiscou et al. (1996) proposed a GA with **row-based crossovers** for D-optimal designs
- Lin (2012) provided a comprehensive review of genetic algorithms in design optimization

Summary



- When the cost of experimentation is **comparable** to the cost of observation, designed experiments should be used
- **Popular** design criteria in SAS/JMP includes D-, A- and E-optimality
- Effective design **construction algorithms** typically employ a combination of exchange algorithms, simulated annealing, particle swarm optimization or genetic algorithms