

# MEM6810 Engineering Systems Modeling and Simulation



## 工程系统建模与仿真

Theory Analysis

### Lecture 10: Output Analysis III: Optimization

SHEN Haihui 沈海辉

Sino-US Global Logistics Institute  
Shanghai Jiao Tong University

 [shenhaihui.github.io/teaching/mem6810f](https://shenhaihui.github.io/teaching/mem6810f)  
 [shenhaihui@sjtu.edu.cn](mailto:shenhaihui@sjtu.edu.cn)

Spring 2022 (full-time)



上海交通大学  
SHANGHAI JIAO TONG UNIVERSITY

董浩云航运与物流研究院

CY TUNG Institute of Maritime and Logistics

中美物流研究院 (工程系统管理研究院)

Sino-US Global Logistics Institute (Institute of Industrial & System Engineering)



- 1 Introduction
  - ▶ Definition
  - ▶ Types
- 2 White-box OvS Problem
  - ▶ Sample Average Approximation
- 3 Black-box COvS Problem
  - ▶ Gradient Descent
  - ▶ Stochastic Approximation
- 4 Black-box DOvS Problem
  - ▶ Simulated Annealing
  - ▶ COMPASS
- 5 Usage in Softwares



- 1 Introduction
  - ▶ Definition
  - ▶ Types
- 2 White-box OvS Problem
  - ▶ Sample Average Approximation
- 3 Black-box COvS Problem
  - ▶ Gradient Descent
  - ▶ Stochastic Approximation
- 4 Black-box DOvS Problem
  - ▶ Simulated Annealing
  - ▶ COMPASS
- 5 Usage in Softwares

- **Optimization via Simulation (OvS)**, or, simply called Simulation Optimization (SO):

$$\min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}) := \mathbb{E}[G(\mathbf{x}, \xi)],$$

where  $\mathcal{X} \subset \mathbb{R}^d$  is the feasible set, and  $g : \mathcal{X} \rightarrow \mathbb{R}$  is a deterministic function whose values can only be evaluated with noisy observations.

- **Optimization via Simulation (OvS)**, or, simply called Simulation Optimization (SO):

$$\min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}) := \mathbb{E}[G(\mathbf{x}, \xi)],$$

where  $\mathcal{X} \subset \mathbb{R}^d$  is the feasible set, and  $g : \mathcal{X} \rightarrow \mathbb{R}$  is a deterministic function whose values can only be evaluated with noisy observations.

- Given  $\mathbf{x}$ ,  $G(\mathbf{x}, \xi)$  is a random variable (the randomness is from  $\xi$ ), and the distribution of  $G(\mathbf{x}, \xi)$  is unknown.

- **Optimization via Simulation (OvS)**, or, simply called Simulation Optimization (SO):

$$\min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}) := \mathbb{E}[G(\mathbf{x}, \xi)],$$

where  $\mathcal{X} \subset \mathbb{R}^d$  is the feasible set, and  $g : \mathcal{X} \rightarrow \mathbb{R}$  is a deterministic function whose values can only be evaluated with noisy observations.

- Given  $\mathbf{x}$ ,  $G(\mathbf{x}, \xi)$  is a random variable (the randomness is from  $\xi$ ), and the distribution of  $G(\mathbf{x}, \xi)$  is unknown.
- Given  $\mathbf{x}$ , realizations of  $G(\mathbf{x}, \xi)$  can be observed by running simulation, or more generally, taking samples.

- OvS Problem can be classified into two types according to whether the explicit form of  $G(\boldsymbol{x}, \xi)$  is available.

- OvS Problem can be classified into two types according to whether the explicit form of  $G(\mathbf{x}, \xi)$  is available.
- **White-box**: The explicit form of  $G(\mathbf{x}, \xi)$  is available.
  - Example:  $G(x, \xi) = \sin((x - \xi)^2)$ , where the distribution of  $\xi$  is unknown.



- OvS Problem can be classified into two types according to whether the explicit form of  $G(\boldsymbol{x}, \xi)$  is available.
- **White-box**: The explicit form of  $G(\boldsymbol{x}, \xi)$  is available.
  - Example:  $G(x, \xi) = \sin((x - \xi)^2)$ , where the distribution of  $\xi$  is unknown.
- **Black-box**: The explicit form of  $G(\boldsymbol{x}, \xi)$  is not available and it is embedded in a simulation model.
  - Example: Let  $G(\boldsymbol{x}, \xi)$  be the waiting time of a customer in a complex queueing network, where  $\boldsymbol{x}$  represents the configuration parameters.

- OvS Problem can be classified into three types according to the feasible set  $\mathcal{X}$ .

- OvS Problem can be classified into three types according to the feasible set  $\mathcal{X}$ .
- **Ranking and selection (R&S)**:  $\mathcal{X}$  is a set of relatively small number of (discrete) solutions.

- OvS Problem can be classified into three types according to the feasible set  $\mathcal{X}$ .
- **Ranking and selection (R&S)**:  $\mathcal{X}$  is a set of relatively small number of (discrete) solutions.
- **Discrete OvS (DOvS)**:  $\mathcal{X}$  is a discrete set, with huge or even countably infinite number of solutions.
  - One can also view R&S problem as a special type of DOvS problem.

- OvS Problem can be classified into three types according to the feasible set  $\mathcal{X}$ .
- **Ranking and selection (R&S)**:  $\mathcal{X}$  is a set of relatively small number of (discrete) solutions.
- **Discrete OvS (DOvS)**:  $\mathcal{X}$  is a discrete set, with huge or even countably infinite number of solutions.
  - One can also view R&S problem as a special type of DOvS problem.
- **Continuous OvS (COvS)**:  $\mathcal{X}$  is a continuous set, hence there exists uncountably infinite number of solutions.

- 1 Introduction
  - ▶ Definition
  - ▶ Types
- 2 White-box OvS Problem
  - ▶ Sample Average Approximation
- 3 Black-box COvS Problem
  - ▶ Gradient Descent
  - ▶ Stochastic Approximation
- 4 Black-box DOvS Problem
  - ▶ Simulated Annealing
  - ▶ COMPASS
- 5 Usage in Softwares



- For white-box OvS problems, we can use the sample average approximation.

- For white-box OvS problems, we can use the sample average approximation.
- Of course, those algorithms designed for black-box OvS problems can also be applied to white-box OvS problems.



- Suppose that we have an iid sample  $\{\xi_1, \dots, \xi_n\}$  of  $\xi$ .
- To solve  $\min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}) := \mathbb{E}[G(\mathbf{x}, \xi)]$ , we try to solve

$$\min_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^n G(\mathbf{x}, \xi_i),$$

with any suitable deterministic optimization algorithm (after  $\{\xi_1, \dots, \xi_n\}$  is realized).

- Suppose that we have an iid sample  $\{\xi_1, \dots, \xi_n\}$  of  $\xi$ .
- To solve  $\min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}) := \mathbb{E}[G(\mathbf{x}, \xi)]$ , we try to solve

$$\min_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^n G(\mathbf{x}, \xi_i),$$

with any suitable deterministic optimization algorithm (after  $\{\xi_1, \dots, \xi_n\}$  is realized).

- This method is called Sample Average Approximation (SAA); see [Kim et al. \(2015\)](#) for a review.

- Suppose that we have an iid sample  $\{\xi_1, \dots, \xi_n\}$  of  $\xi$ .
- To solve  $\min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}) := \mathbb{E}[G(\mathbf{x}, \xi)]$ , we try to solve

$$\min_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) := \frac{1}{n} \sum_{i=1}^n G(\mathbf{x}, \xi_i),$$

with any suitable deterministic optimization algorithm (after  $\{\xi_1, \dots, \xi_n\}$  is realized).

- This method is called Sample Average Approximation (SAA); see [Kim et al. \(2015\)](#) for a review.
- Clearly, for finite  $n$ ,  $\inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x})$  is a random variable (before  $\{\xi_1, \dots, \xi_n\}$  is realized), and it is not strictly equal to  $\min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x})$ .

- Indeed, one can prove that

$$\mathbb{E} \left[ \inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \right] \leq \min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}).$$

- Indeed, one can prove that

$$\mathbb{E} \left[ \inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \right] \leq \min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}).$$

Proof.

- Indeed, one can prove that

$$\mathbb{E} \left[ \inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \right] \leq \min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}).$$

*Proof.* For any  $\mathbf{y} \in \mathcal{X}$ ,

$$\inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \leq \hat{g}_n(\mathbf{y}) \implies \mathbb{E} \left[ \inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \right] \leq \mathbb{E}[\hat{g}_n(\mathbf{y})] = g(\mathbf{y}).$$

- Indeed, one can prove that

$$\mathbb{E} \left[ \inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \right] \leq \min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}).$$

*Proof.* For any  $\mathbf{y} \in \mathcal{X}$ ,

$$\inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \leq \hat{g}_n(\mathbf{y}) \implies \mathbb{E} \left[ \inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \right] \leq \mathbb{E}[\hat{g}_n(\mathbf{y})] = g(\mathbf{y}).$$

Minimizing the right-hand side over all  $\mathbf{y} \in \mathcal{X}$  completes the proof. ■

- Indeed, one can prove that

$$\mathbb{E} \left[ \inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \right] \leq \min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}).$$

Proof. For any  $\mathbf{y} \in \mathcal{X}$ ,

$$\inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \leq \hat{g}_n(\mathbf{y}) \implies \mathbb{E} \left[ \inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \right] \leq \mathbb{E}[\hat{g}_n(\mathbf{y})] = g(\mathbf{y}).$$

Minimizing the right-hand side over all  $\mathbf{y} \in \mathcal{X}$  completes the proof. ■

- Moreover, it can also be shown that

$$\mathbb{E} \left[ \inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_n(\mathbf{x}) \right] \leq \mathbb{E} \left[ \inf_{\mathbf{x} \in \mathcal{X}} \hat{g}_{n+1}(\mathbf{x}) \right] \leq \min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}).$$

(Prove it as an exercise)



- What can we say if we continuously increase sample size  $n$ ?

- What can we say if we continuously increase sample size  $n$ ?
- It will be **reassuring** if we know that the obtained solution will be closer and closer to the true solution, as we increase sample size  $n$ .

- What can we say if we continuously increase sample size  $n$ ?
- It will be **reassuring** if we know that the obtained solution will be closer and closer to the true solution, as we increase sample size  $n$ .
- Formally, we are seeking for a **convergence** guarantee for SAA method.

- For set  $\mathcal{A} \subset \mathbb{R}^d$ , the distance from  $\mathbf{x} \in \mathbb{R}^d$  to  $\mathcal{A}$  is defined as

$$\text{dist}(\mathbf{x}, \mathcal{A}) := \inf_{\mathbf{y} \in \mathcal{A}} \|\mathbf{x} - \mathbf{y}\|,$$

where  $\|\cdot\|$  denotes the Euclidean distance.

- For set  $\mathcal{A} \subset \mathbb{R}^d$ , the distance from  $\mathbf{x} \in \mathbb{R}^d$  to  $\mathcal{A}$  is defined as

$$\text{dist}(\mathbf{x}, \mathcal{A}) := \inf_{\mathbf{y} \in \mathcal{A}} \|\mathbf{x} - \mathbf{y}\|,$$

where  $\|\cdot\|$  denotes the Euclidean distance.

- For sets  $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^d$ , the deviation from  $\mathcal{A}$  to  $\mathcal{B}$  is defined as

$$D(\mathcal{A}, \mathcal{B}) := \sup_{\mathbf{x} \in \mathcal{A}} \text{dist}(\mathbf{x}, \mathcal{B}).$$

- For set  $\mathcal{A} \subset \mathbb{R}^d$ , the distance from  $\mathbf{x} \in \mathbb{R}^d$  to  $\mathcal{A}$  is defined as

$$\text{dist}(\mathbf{x}, \mathcal{A}) := \inf_{\mathbf{y} \in \mathcal{A}} \|\mathbf{x} - \mathbf{y}\|,$$

where  $\|\cdot\|$  denotes the Euclidean distance.

- For sets  $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^d$ , the deviation from  $\mathcal{A}$  to  $\mathcal{B}$  is defined as

$$D(\mathcal{A}, \mathcal{B}) := \sup_{\mathbf{x} \in \mathcal{A}} \text{dist}(\mathbf{x}, \mathcal{B}).$$

- Let

$$\mathcal{S} := \underset{\mathbf{x} \in \mathcal{X}}{\text{argmin}} g(\mathbf{x}),$$

$$\widehat{\mathcal{S}}_n := \underset{\mathbf{x} \in \mathcal{X}}{\text{argmin}} \widehat{g}_n(\mathbf{x}).$$

## Convergence of SAA (Theorem 5.3 of Shapiro et al. (2009))

Suppose that

- 1  $\mathcal{X}$  is a compact set;
- 2  $g(x)$  is finite valued and continuous on  $\mathcal{X}$ ;
- 3  $\mathbb{P}\{\widehat{g}_n(\mathbf{x}) \rightarrow g(\mathbf{x}) \text{ uniformly in } \mathbf{x} \in \mathcal{X}\} = 1$ ;
- 4  $\mathbb{P}\{\widehat{\mathcal{S}}_n \text{ is nonempty for } n \text{ large enough}\} = 1$ ;

Then, as  $n \rightarrow \infty$ ,

$$\min_{\mathbf{x} \in \mathcal{X}} \widehat{g}_n(\mathbf{x}) \xrightarrow{a.s.} \min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}), \text{ and } D(\widehat{\mathcal{S}}_n, \mathcal{S}) \xrightarrow{a.s.} 0.$$

## Convergence of SAA (Theorem 5.3 of Shapiro et al. (2009))

Suppose that

- 1  $\mathcal{X}$  is a compact set;
- 2  $g(x)$  is finite valued and continuous on  $\mathcal{X}$ ;
- 3  $\mathbb{P}\{\widehat{g}_n(x) \rightarrow g(x) \text{ uniformly in } x \in \mathcal{X}\} = 1$ ;
- 4  $\mathbb{P}\{\widehat{\mathcal{S}}_n \text{ is nonempty for } n \text{ large enough}\} = 1$ ;

Then, as  $n \rightarrow \infty$ ,

$$\min_{x \in \mathcal{X}} \widehat{g}_n(x) \xrightarrow{a.s.} \min_{x \in \mathcal{X}} g(x), \text{ and } D(\widehat{\mathcal{S}}_n, \mathcal{S}) \xrightarrow{a.s.} 0.$$

Besides, if  $\mathcal{S} = \{\mathbf{x}^*\}$  is a singleton, then for any  $\widehat{\mathbf{x}}_n \in \widehat{\mathcal{S}}_n$ ,

$$\widehat{\mathbf{x}}_n \xrightarrow{a.s.} \mathbf{x}^*, \text{ as } n \rightarrow \infty.$$



- **How fast** does the SAA solution converge to the true solution?

- **How fast** does the SAA solution converge to the true solution?
- Formally, it's known as the **rate of convergence**.

- **How fast** does the SAA solution converge to the true solution?
- Formally, it's known as the **rate of convergence**.
- Under certain regularity conditions, one may show that

$$\left| \min_{\mathbf{x} \in \mathcal{X}} \widehat{g}_n(\mathbf{x}) - \min_{\mathbf{x} \in \mathcal{X}} g(\mathbf{x}) \right| = O_p(n^{-1/2}),$$

and given  $\mathcal{S} = \{\mathbf{x}^*\}$  is a singleton,

$$\|\widehat{\mathbf{x}}_n - \mathbf{x}^*\| = O_p(n^{-1/2}).$$

- 1 Introduction
  - ▶ Definition
  - ▶ Types
- 2 White-box OvS Problem
  - ▶ Sample Average Approximation
- 3 Black-box COvS Problem
  - ▶ Gradient Descent
  - ▶ Stochastic Approximation
- 4 Black-box DOvS Problem
  - ▶ Simulated Annealing
  - ▶ COMPASS
- 5 Usage in Softwares



- Main types of algorithms for black-box COvS problems:
  - random search; see [Andradóttir \(2015\)](#) for a review;
  - stochastic approximation; see [Chau and Fu \(2015\)](#) for a review;
  - surrogate-based methods; see [Hong and Zhang \(2021\)](#) for a review.

- Main types of algorithms for black-box COvS problems:
  - random search; see [Andradóttir \(2015\)](#) for a review;
  - stochastic approximation; see [Chau and Fu \(2015\)](#) for a review;
  - surrogate-based methods; see [Hong and Zhang \(2021\)](#) for a review.
- Stochastic Approximation (SA) was proposed by [Robbins and Monro \(1951\)](#) and [Kiefer and Wolfowitz \(1952\)](#).
- SA can be viewed as a stochastic version of the gradient descent (or called steepest descent) algorithm, so it is also called stochastic gradient descent.

- Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable (deterministic) function:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma \nabla g(\mathbf{x}_k),$$

where  $\nabla g(\mathbf{x})$  is the gradient and  $\gamma > 0$  is the step size.

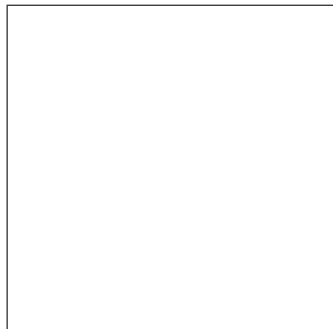
- Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable (deterministic) function:

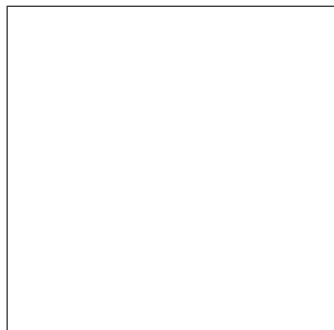
$$\mathbf{x}_{k+1} = \mathbf{x}_k - \gamma \nabla g(\mathbf{x}_k),$$

where  $\nabla g(\mathbf{x})$  is the gradient and  $\gamma > 0$  is the step size.

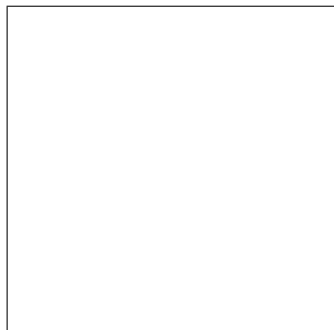
- If the minimization problem is constrained, say the feasible set  $\mathcal{X} \subset \mathbb{R}^d$  is convex and compact, one can easily add a projection  $\Pi_{\mathcal{X}}(\mathbf{x})$  mapping  $\mathbf{x} \notin \mathcal{X}$  back into  $\mathcal{X}$ .







- The value of the step size  $\gamma$  is allowed to change at every iteration, and with proper choice, convergence to a local minimizer (say,  $\mathbf{x}^*$ ) can be guaranteed, i.e.,  $\mathbf{x}_k \rightarrow \mathbf{x}^*$ .



- The value of the step size  $\gamma$  is allowed to change at every iteration, and with proper choice, convergence to a local minimizer (say,  $\mathbf{x}^*$ ) can be guaranteed, i.e.,  $\mathbf{x}_k \rightarrow \mathbf{x}^*$ .
- Under certain regularity conditions, one can show that  $|g(\mathbf{x}_k) - g(\mathbf{x}^*)| = O(k^{-1})$  for unconstrained problem with constant  $\gamma$ .

- SA as a stochastic version of the gradient ascent:

$$\mathbf{X}_{k+1} = \Pi_{\mathcal{X}} \left( \mathbf{X}_k - a_k \widehat{\nabla} g(\mathbf{X}_k) \right),$$

where  $\Pi_{\mathcal{X}}$  is the projection,  $\{a_k\}_{k \geq 1}$  is a deterministic positive sequence for step size, and  $\widehat{\nabla} g(\mathbf{x})$  is an estimator of the gradient  $\nabla g(\mathbf{x})$ .

- SA as a stochastic version of the gradient ascent:

$$\mathbf{X}_{k+1} = \Pi_{\mathcal{X}} \left( \mathbf{X}_k - a_k \widehat{\nabla} g(\mathbf{X}_k) \right),$$

where  $\Pi_{\mathcal{X}}$  is the projection,  $\{a_k\}_{k \geq 1}$  is a deterministic positive sequence for step size, and  $\widehat{\nabla} g(\mathbf{x})$  is an estimator of the gradient  $\nabla g(\mathbf{x})$ .

- In some simulation experiments, unbiased  $\widehat{\nabla} g(\mathbf{x})$  is available,<sup>†</sup> then it is the Robbins-Monro (RM) type SA (Robbins and Monro 1951).

---

<sup>†</sup>When we observe  $G(\mathbf{x}, \xi)$ , we will also observe  $\widehat{\nabla} g(\mathbf{x}, \xi)$  at the same time such that  $\mathbb{E}[\widehat{\nabla} g(\mathbf{x}, \xi)] = \nabla g(\mathbf{x})$ .

- SA as a stochastic version of the gradient ascent:

$$\mathbf{X}_{k+1} = \Pi_{\mathcal{X}} \left( \mathbf{X}_k - a_k \widehat{\nabla} g(\mathbf{X}_k) \right),$$

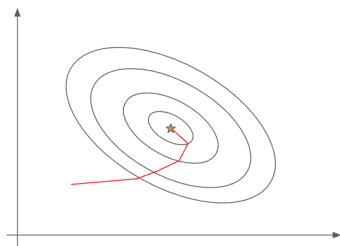
where  $\Pi_{\mathcal{X}}$  is the projection,  $\{a_k\}_{k \geq 1}$  is a deterministic positive sequence for step size, and  $\widehat{\nabla} g(\mathbf{x})$  is an estimator of the gradient  $\nabla g(\mathbf{x})$ .

- In some simulation experiments, unbiased  $\widehat{\nabla} g(\mathbf{x})$  is available,<sup>†</sup> then it is the Robbins-Monro (RM) type SA (Robbins and Monro 1951).
- Otherwise,  $\widehat{\nabla} g(\mathbf{x})$  needs to be constructed with certain indirect method (thus biased), then it is the Kiefer-Wolfowitz (KW) type SA (Kiefer and Wolfowitz (1952)).

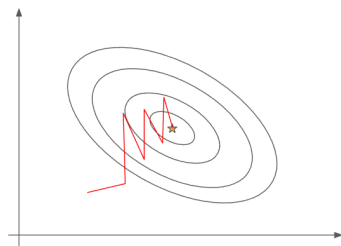
---

<sup>†</sup> When we observe  $G(\mathbf{x}, \xi)$ , we will also observe  $\widehat{\nabla} g(\mathbf{x}, \xi)$  at the same time such that  $\mathbb{E}[\widehat{\nabla} g(\mathbf{x}, \xi)] = \nabla g(\mathbf{x})$ .

- Gradient descent vs SA (i.e., stochastic gradient descent):



Gradient Descent



Stochastic Gradient Descent



- Construct  $\widehat{\nabla}g(\mathbf{X}_k)$  via symmetric (or central) finite difference:

$$\widehat{\nabla}g(\mathbf{X}_k) := (g_1(\mathbf{X}_k), \dots, g_d(\mathbf{X}_k))^{\top},$$

where

$$g_i(\mathbf{X}_k) := \frac{G(\mathbf{X}_k + c_k \mathbf{e}_i) - G(\mathbf{X}_k - c_k \mathbf{e}_i)}{2c_k},$$

$\mathbf{e}_i$  denotes a  $d \times 1$  vector whose  $i$ th element is one and other elements are all zeros,  $i = 1, \dots, d$ , and  $\{c_k\}_{k \geq 1}$  is a deterministic positive sequence.



- Construct  $\widehat{\nabla}g(\mathbf{X}_k)$  via symmetric (or central) finite difference:

$$\widehat{\nabla}g(\mathbf{X}_k) := (g_1(\mathbf{X}_k), \dots, g_d(\mathbf{X}_k))^{\top},$$

where

$$g_i(\mathbf{X}_k) := \frac{G(\mathbf{X}_k + c_k \mathbf{e}_i) - G(\mathbf{X}_k - c_k \mathbf{e}_i)}{2c_k},$$

$\mathbf{e}_i$  denotes a  $d \times 1$  vector whose  $i$ th element is one and other elements are all zeros,  $i = 1, \dots, d$ , and  $\{c_k\}_{k \geq 1}$  is a deterministic positive sequence.

- It requires  $2d$  **additional** simulation runs (samples) to compute  $\widehat{\nabla}g(\mathbf{X}_k)$ .

- Let  $\mathcal{M}$  denote the set of local optimal solutions:

$$\mathcal{M} := \left\{ \mathbf{x} \in \mathcal{X} : g(\mathbf{x}) \leq \min_{\mathbf{y} \in \mathcal{B}(\mathbf{x})} g(\mathbf{y}) \right\},$$

where  $\mathcal{B}(\mathbf{x}) \subset \mathcal{X}$  denotes a neighborhood of  $\mathbf{x} \in \mathcal{X}$ .

- Let  $\mathcal{M}$  denote the set of local optimal solutions:

$$\mathcal{M} := \left\{ \mathbf{x} \in \mathcal{X} : g(\mathbf{x}) \leq \min_{\mathbf{y} \in \mathcal{B}(\mathbf{x})} g(\mathbf{y}) \right\},$$

where  $\mathcal{B}(\mathbf{x}) \subset \mathcal{X}$  denotes a neighborhood of  $\mathbf{x} \in \mathcal{X}$ .

### Local Convergence of SA (Theorem 3 of Blum (1954))

Suppose that

- 1  $g(x)$  satisfies certain regularity conditions;
- 2  $\text{Var}(G(\mathbf{x}, \xi)) \leq \sigma^2 < \infty$ ;
- 3  $\lim_{k \rightarrow \infty} c_k = 0$ ,  $\sum_{k=1}^{\infty} a_k = \infty$ ,  $\sum_{k=1}^{\infty} a_k c_k < \infty$ , and  $\sum_{k=1}^{\infty} a_k^2 c_k^{-2} < \infty$ .

Then, for KW type SA with symmetric difference gradient estimator,  $\text{dist}(\mathbf{X}_k, \mathcal{M}) \xrightarrow{a.s.} 0$  as  $k \rightarrow \infty$ .

- Under certain conditions, for  $\mathbf{x}^* \in \mathcal{M}$  such that  $\mathbf{X}_k \xrightarrow{a.s.} \mathbf{x}^*$ , RM type SA can reach  $O_p(k^{-1/2})$  rate of convergence, i.e.,

$$\|\mathbf{X}_k - \mathbf{x}^*\| = O_p(k^{-1/2}),$$

while KW type SA can reach  $O_p(k^{-1/3})$  rate of convergence.

- Under certain conditions, for  $\mathbf{x}^* \in \mathcal{M}$  such that  $\mathbf{X}_k \xrightarrow{a.s.} \mathbf{x}^*$ , RM type SA can reach  $O_p(k^{-1/2})$  rate of convergence, i.e.,

$$\|\mathbf{X}_k - \mathbf{x}^*\| = O_p(k^{-1/2}),$$

while KW type SA can reach  $O_p(k^{-1/3})$  rate of convergence.

- Note that the above order is in terms of the iteration number  $k$ , rather than the number of simulation runs (sample size).

- Under certain conditions, for  $\mathbf{x}^* \in \mathcal{M}$  such that  $\mathbf{X}_k \xrightarrow{a.s.} \mathbf{x}^*$ , RM type SA can reach  $O_p(k^{-1/2})$  rate of convergence, i.e.,

$$\|\mathbf{X}_k - \mathbf{x}^*\| = O_p(k^{-1/2}),$$

while KW type SA can reach  $O_p(k^{-1/3})$  rate of convergence.

- Note that the above order is in terms of the iteration number  $k$ , rather than the number of simulation runs (sample size).
- If in terms of the sample size  $n$ , the rate of convergence of KW type SA is  $O_p((n/d)^{-1/3})$ , which depends on the dimensionality  $d$ .

- Simultaneous perturbation stochastic approximation (SPSA):

$$\widehat{\nabla} g(\mathbf{X}_k) := (g_1(\mathbf{X}_k), \dots, g_d(\mathbf{X}_k))^\top,$$

where

$$g_i(\mathbf{X}_k) := \frac{G(\mathbf{X}_k + c_k \mathbf{B}_k) - G(\mathbf{X}_k - c_k \mathbf{B}_k)}{2c_k B_{k,i}},$$

$\mathbf{B}_k := (B_{k,1}, \dots, B_{k,d})^\top$ , and  $B_{k,i} = 1$  or  $-1$  with probability  $1/2$ .

- Simultaneous perturbation stochastic approximation (SPSA):

$$\widehat{\nabla} g(\mathbf{X}_k) := (g_1(\mathbf{X}_k), \dots, g_d(\mathbf{X}_k))^\top,$$

where

$$g_i(\mathbf{X}_k) := \frac{G(\mathbf{X}_k + c_k \mathbf{B}_k) - G(\mathbf{X}_k - c_k \mathbf{B}_k)}{2c_k B_{k,i}},$$

$\mathbf{B}_k := (B_{k,1}, \dots, B_{k,d})^\top$ , and  $B_{k,i} = 1$  or  $-1$  with probability  $1/2$ .

- It requires only 2 **additional** simulation runs (samples) to compute  $\widehat{\nabla} g(\mathbf{X}_k)$ , no matter what  $d$  is.



- Simultaneous perturbation stochastic approximation (SPSA):

$$\widehat{\nabla} g(\mathbf{X}_k) := (g_1(\mathbf{X}_k), \dots, g_d(\mathbf{X}_k))^\top,$$

where

$$g_i(\mathbf{X}_k) := \frac{G(\mathbf{X}_k + c_k \mathbf{B}_k) - G(\mathbf{X}_k - c_k \mathbf{B}_k)}{2c_k B_{k,i}},$$

$\mathbf{B}_k := (B_{k,1}, \dots, B_{k,d})^\top$ , and  $B_{k,i} = 1$  or  $-1$  with probability  $1/2$ .

- It requires only 2 **additional** simulation runs (samples) to compute  $\widehat{\nabla} g(\mathbf{X}_k)$ , no matter what  $d$  is.
- SPSA can reach  $O_p(n^{-1/3})$  rate of convergence in terms of the sample size  $n$ .

- 1 Introduction
  - ▶ Definition
  - ▶ Types
- 2 White-box OvS Problem
  - ▶ Sample Average Approximation
- 3 Black-box COvS Problem
  - ▶ Gradient Descent
  - ▶ Stochastic Approximation
- 4 Black-box DOvS Problem
  - ▶ Simulated Annealing
  - ▶ COMPASS
- 5 Usage in Softwares



- Many black-box DOvS algorithms are based on random search; see [Hong et al. \(2015\)](#) for a review.

- Many black-box DOvS algorithms are based on random search; see [Hong et al. \(2015\)](#) for a review.
- The framework of random search:
  - Initialization:
  - At Iteration  $k$ :
    - Sampling:
    - Evaluation:
    - Updating:

- Many black-box DOvS algorithms are based on random search; see [Hong et al. \(2015\)](#) for a review.
- The framework of random search:
  - **Initialization:** Arbitrarily choose  $x_0^* \in \mathcal{X}$ ; set the information set (that keeps visited solutions and their corresponding observations)  $\mathcal{F}_0$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:**
    - **Evaluation:**
    - **Updating:**

- Many black-box DOvS algorithms are based on random search; see [Hong et al. \(2015\)](#) for a review.
- The framework of random search:
  - **Initialization:** Arbitrarily choose  $x_0^* \in \mathcal{X}$ ; set the information set (that keeps visited solutions and their corresponding observations)  $\mathcal{F}_0$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:** Choose the estimation set  $\mathcal{E} \subset \mathcal{X}$  (that contains solutions at which simulation will be run); some or all of the solutions in  $\mathcal{E}$  are randomly sampled from  $\mathcal{X}$  with distribution determined by information  $\mathcal{F}_k$ .
    - **Evaluation:**
    - **Updating:**

- Many black-box DOvS algorithms are based on random search; see [Hong et al. \(2015\)](#) for a review.
- The framework of random search:
  - **Initialization:** Arbitrarily choose  $x_0^* \in \mathcal{X}$ ; set the information set (that keeps visited solutions and their corresponding observations)  $\mathcal{F}_0$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:** Choose the estimation set  $\mathcal{E} \subset \mathcal{X}$  (that contains solutions at which simulation will be run); some or all of the solutions in  $\mathcal{E}$  are randomly sampled from  $\mathcal{X}$  with distribution determined by information  $\mathcal{F}_k$ .
    - **Evaluation:** For each  $x \in \mathcal{E}$ , spend simulation effort according to certain rule determined by  $\mathcal{F}_k$  and  $\mathcal{E}$ .
    - **Updating:**

- Many black-box DOvS algorithms are based on random search; see [Hong et al. \(2015\)](#) for a review.
- The framework of random search:
  - **Initialization:** Arbitrarily choose  $x_0^* \in \mathcal{X}$ ; set the information set (that keeps visited solutions and their corresponding observations)  $\mathcal{F}_0$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:** Choose the estimation set  $\mathcal{E} \subset \mathcal{X}$  (that contains solutions at which simulation will be run); some or all of the solutions in  $\mathcal{E}$  are randomly sampled from  $\mathcal{X}$  with distribution determined by information  $\mathcal{F}_k$ .
    - **Evaluation:** For each  $x \in \mathcal{E}$ , spend simulation effort according to certain rule determined by  $\mathcal{F}_k$  and  $\mathcal{E}$ .
    - **Updating:** Update  $\mathcal{F}_{k+1}$ ; choose some  $x_{k+1}^*$  as the current best solution based on certain estimator; set  $k \leftarrow k + 1$ .



- The simulated annealing algorithm dates back to the pioneering work by [Metropolis et al. \(1953\)](#).
  - It studied how in the physical annealing process, particles of a solid arrange themselves into thermal equilibrium at a given temperature.

- The simulated annealing algorithm dates back to the pioneering work by [Metropolis et al. \(1953\)](#).
  - It studied how in the physical annealing process, particles of a solid arrange themselves into thermal equilibrium at a given temperature.
- A large body of literature has developed the simulated annealing algorithm to solve deterministic global optimization problems over **finite** set; important works include [Kirkpatrick et al. \(1983\)](#), [Mitra et al. \(1986\)](#), [Hajek \(1988\)](#), etc.

- The simulated annealing algorithm dates back to the pioneering work by [Metropolis et al. \(1953\)](#).
  - It studied how in the physical annealing process, particles of a solid arrange themselves into thermal equilibrium at a given temperature.
- A large body of literature has developed the simulated annealing algorithm to solve deterministic global optimization problems over **finite** set; important works include [Kirkpatrick et al. \(1983\)](#), [Mitra et al. \(1986\)](#), [Hajek \(1988\)](#), etc.
- Later, the simulated annealing was extended to solve black-box DOvS problems over **finite** set; important works include [Bulgak and Sander \(1988\)](#), [Gelfand and Mitter \(1989\)](#), [Alrefaei and Andradóttir \(1999\)](#), etc.

- Let  $\mathcal{B}(\mathbf{x}) \subset \mathcal{X}$  denote a neighborhood<sup>†</sup> of  $\mathbf{x} \in \mathcal{X}$ .

---

<sup>†</sup>The neighborhood structure can be quite different in discrete optimization compared to continuous optimization!

- Let  $\mathcal{B}(\mathbf{x}) \subset \mathcal{X}$  denote a neighborhood<sup>†</sup> of  $\mathbf{x} \in \mathcal{X}$ .
- $\mathcal{B}(\mathbf{x})$  is carefully desined such that, for any  $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ ,  $\mathbf{y}$  is reachable from  $\mathbf{x}$ .
  - That is, there exists a finite sequence  $\mathbf{x} = \mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_\ell = \mathbf{y}$  such that  $\mathbf{x}_{i+1} \in \mathcal{B}(\mathbf{x}_i)$ ,  $i = 0, 1, \dots, \ell - 1$ .

---

<sup>†</sup>The neighborhood structer can be quite different in discrete optimization compared to continuous optimization!

- Let  $\mathcal{B}(x) \subset \mathcal{X}$  denote a neighborhood<sup>†</sup> of  $x \in \mathcal{X}$ .
- $\mathcal{B}(x)$  is carefully desined such that, for any  $x, y \in \mathcal{X}$ ,  $y$  is reachable from  $x$ .
  - That is, there exists a finite sequence  $x = x_0, x_1, \dots, x_\ell = y$  such that  $x_{i+1} \in \mathcal{B}(x_i)$ ,  $i = 0, 1, \dots, \ell - 1$ .
- Define transition probability  $R(x, y)$ , where  $R : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty)$  and  $R(x, y) > 0 \iff y \in \mathcal{B}(x)$ .

---

<sup>†</sup>The neighborhood structer can be quite different in discrete optimization compared to continuous optimization!

- Let  $\mathcal{B}(x) \subset \mathcal{X}$  denote a neighborhood<sup>†</sup> of  $x \in \mathcal{X}$ .
- $\mathcal{B}(x)$  is carefully desined such that, for any  $x, y \in \mathcal{X}$ ,  $y$  is reachable from  $x$ .
  - That is, there exists a finite sequence  $x = x_0, x_1, \dots, x_\ell = y$  such that  $x_{i+1} \in \mathcal{B}(x_i)$ ,  $i = 0, 1, \dots, \ell - 1$ .
- Define transition probability  $R(x, y)$ , where  $R: \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty)$  and  $R(x, y) > 0 \iff y \in \mathcal{B}(x)$ .
- Let  $\{t_k\}_{k \geq 1}$  be a positive sequence of numbers, which is konwn as the temperature.

---

<sup>†</sup>The neighborhood structer can be quite different in discrete optimization compared to continuous optimization!

- Simulated annealing algorithm for deterministic optimization:
  - Initialization:
  - At Iteration  $k$ :
    - Sampling:
    - Evaluation: No need in the deterministic optimization.
    - Updating:



- Simulated annealing algorithm for deterministic optimization:
  - **Initialization:** Arbitrarily choose  $\mathbf{X}_0 \in \mathcal{X}$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:**
    - **Evaluation:** No need in the deterministic optimization.
    - **Updating:**

- Simulated annealing algorithm for deterministic optimization:
  - **Initialization:** Arbitrarily choose  $\mathbf{X}_0 \in \mathcal{X}$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:** Sample a candidate solution  $\mathbf{Y}_{k+1} \in \mathcal{B}(\mathbf{X}_k)$  according to distribution  $R(\mathbf{X}_k, \cdot)$ , i.e.,

$$\mathbb{P}(\mathbf{Y}_{k+1} = \mathbf{y} | \mathbf{X}_k = \mathbf{x}) = R(\mathbf{x}, \mathbf{y}).$$

- **Evaluation:** No need in the deterministic optimization.
- **Updating:**

- Simulated annealing algorithm for deterministic optimization:
  - **Initialization:** Arbitrarily choose  $\mathbf{X}_0 \in \mathcal{X}$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:** Sample a candidate solution  $\mathbf{Y}_{k+1} \in \mathcal{B}(\mathbf{X}_k)$  according to distribution  $R(\mathbf{X}_k, \cdot)$ , i.e.,

$$\mathbb{P}(\mathbf{Y}_{k+1} = \mathbf{y} | \mathbf{X}_k = \mathbf{x}) = R(\mathbf{x}, \mathbf{y}).$$

- **Evaluation:** No need in the deterministic optimization.
- **Updating:** Let

$$\mathbf{X}_{k+1} := \begin{cases} \mathbf{Y}_{k+1}, & \text{with probability } \exp\left\{\frac{-[g(\mathbf{Y}_{k+1}) - g(\mathbf{X}_k)]^+}{t_{k+1}}\right\}, \\ \mathbf{X}_k, & \text{otherwise;} \end{cases}$$

set  $k \leftarrow k + 1$ .

- To ensure the simulated annealing algorithm for deterministic optimization is globally convergent, i.e.,

$$\text{dist}(\mathbf{X}_k, \mathcal{S}) \xrightarrow{a.s.} 0, \text{ as } k \rightarrow \infty ,$$

Hajek (1988, Theorem 1) gives a sufficient condition.

- To ensure the simulated annealing algorithm for deterministic optimization is globally convergent, i.e.,

$$\text{dist}(\mathbf{X}_k, \mathcal{S}) \xrightarrow{a.s.} 0, \text{ as } k \rightarrow \infty,$$

Hajek (1988, Theorem 1) gives a sufficient condition.

- $R(\mathbf{x}, \mathbf{y})$  satisfies weak reversibility; a sufficient example is that

$$R(\mathbf{x}, \mathbf{y}) := \begin{cases} \frac{1}{|\mathcal{B}(\mathbf{x})|}, & \text{if } \mathbf{y} \in \mathcal{B}(\mathbf{x}), \\ 0, & \text{otherwise,} \end{cases}$$

with symmetric neighborhood, i.e.,  $\mathbf{y} \in \mathcal{B}(\mathbf{x}) \iff \mathbf{x} \in \mathcal{B}(\mathbf{y})$ .

- To ensure the simulated annealing algorithm for deterministic optimization is globally convergent, i.e.,

$$\text{dist}(\mathbf{X}_k, \mathcal{S}) \xrightarrow{a.s.} 0, \text{ as } k \rightarrow \infty,$$

Hajek (1988, Theorem 1) gives a sufficient condition.

- $R(\mathbf{x}, \mathbf{y})$  satisfies weak reversibility; a sufficient example is that

$$R(\mathbf{x}, \mathbf{y}) := \begin{cases} \frac{1}{|\mathcal{B}(\mathbf{x})|}, & \text{if } \mathbf{y} \in \mathcal{B}(\mathbf{x}), \\ 0, & \text{otherwise,} \end{cases}$$

with symmetric neighborhood, i.e.,  $\mathbf{y} \in \mathcal{B}(\mathbf{x}) \iff \mathbf{x} \in \mathcal{B}(\mathbf{y})$ .

- $\{t_k\}_{k \geq 1}$  takes the form

$$t_k = \frac{c}{\ln(k+1)},$$

where  $c$  is sufficiently large.<sup>†</sup>

<sup>†</sup>  $c \geq d^*$ , where  $d^*$  is the maximum depth (Hajek (1988, p313)) of the local but not global optimal solutions.

- Simulated annealing algorithm for black-box DOvS (Gelfand and Mitter 1989):

- **Initialization:** Arbitrarily choose  $\mathbf{X}_0 \in \mathcal{X}$ ; set iteration index  $k = 0$ .
- **At Iteration  $k$ :**
  - **Sampling:** Sample a candidate solution  $\mathbf{Y}_{k+1} \in \mathcal{B}(\mathbf{X}_k)$  according to distribution  $R(\mathbf{X}_k, \cdot)$ , i.e.,

$$\mathbb{P}(\mathbf{Y}_{k+1} = \mathbf{y} | \mathbf{X}_k = \mathbf{x}) = R(\mathbf{x}, \mathbf{y}).$$

- **Evaluation:** Let  $\hat{g}(\mathbf{Y}_{k+1}) := \frac{1}{n_{k+1}} \sum_{i=1}^{n_{k+1}} G(\mathbf{Y}_{k+1}, \xi_i)$ ,  
 $\hat{g}(\mathbf{X}_k) := \frac{1}{n_{k+1}} \sum_{i=1}^{n_{k+1}} G(\mathbf{X}_k, \xi_i)$ .<sup>†</sup>
- **Updating:** Let

$$\mathbf{X}_{k+1} := \begin{cases} \mathbf{Y}_{k+1}, & \text{with probability } \exp\left\{\frac{-[\hat{g}(\mathbf{Y}_{k+1}) - \hat{g}(\mathbf{X}_k)]^+}{t_{k+1}}\right\}, \\ \mathbf{X}_k, & \text{otherwise;} \end{cases}$$

set  $k \leftarrow k + 1$ .

<sup>†</sup>  $\xi_i$  in  $G(\mathbf{Y}_{k+1}, \xi_i)$  and  $\xi_i$  in  $G(\mathbf{X}_k, \xi_i)$  denote different randomness; written in this way just for notation simplicity.

- Gelfand and Mitter (1989) show that if

$$\hat{g}(\mathbf{Y}_{k+1}) | \mathbf{Y}_{k+1} = \mathbf{y} \sim \mathcal{N}(g(\mathbf{y}), \sigma_{k+1}^2),$$

such that  $\sigma_k = o(t_k)$ , then the simulated annealing algorithm used for DOvS has the same global convergence as its counterpart used for deterministic optimization.



- Gelfand and Mitter (1989) show that if

$$\widehat{g}(\mathbf{Y}_{k+1}) | \mathbf{Y}_{k+1} = \mathbf{y} \sim \mathcal{N}(g(\mathbf{y}), \sigma_{k+1}^2),$$

such that  $\sigma_k = o(t_k)$ , then the simulated annealing algorithm used for DOvS has the same global convergence as its counterpart used for deterministic optimization.

- A sufficient condition is that:
  - $G(\mathbf{x}, \xi) \sim \mathcal{N}(g(\mathbf{x}), \sigma^2(\mathbf{x}))$  with  $\sigma^2(\mathbf{x}) \leq \sigma^2 < \infty$  for all  $\mathbf{x} \in \mathcal{X}$ .
  - $\{n_k\}_{k \geq 1}$  satisfies  $\lim_{k \rightarrow \infty} \frac{1}{t_k \sqrt{n_k}} = 0$ , i.e.,  $n_k := t_k^{-\alpha}$  with  $\alpha > 2$ .

- Gelfand and Mitter (1989) show that if

$$\widehat{g}(\mathbf{Y}_{k+1}) | \mathbf{Y}_{k+1} = \mathbf{y} \sim \mathcal{N}(g(\mathbf{y}), \sigma_{k+1}^2),$$

such that  $\sigma_k = o(t_k)$ , then the simulated annealing algorithm used for DOvS has the same global convergence as its counterpart used for deterministic optimization.

- A sufficient condition is that:
  - $G(\mathbf{x}, \xi) \sim \mathcal{N}(g(\mathbf{x}), \sigma^2(\mathbf{x}))$  with  $\sigma^2(\mathbf{x}) \leq \sigma^2 < \infty$  for all  $\mathbf{x} \in \mathcal{X}$ .
  - $\{n_k\}_{k \geq 1}$  satisfies  $\lim_{k \rightarrow \infty} \frac{1}{t_k \sqrt{n_k}} = 0$ , i.e.,  $n_k := t_k^{-\alpha}$  with  $\alpha > 2$ .
- Alrefaei and Andradóttir (1999) propose a modified simulated annealing algorithm for DOvS, which is also globally convergent:
  - temperature  $t_k$  is constant;
  - the current best solution is chosen in a different way.

- Convergent Optimization via Most-Promising-Area Stochastic Search (COMPASS) is a **locally convergent** algorithm for black-box algorithm proposed by [Hong and Nelson \(2006\)](#).

- Convergent Optimization via Most-Promising-Area Stochastic Search (COMPASS) is a **locally convergent** algorithm for black-box algorithm proposed by [Hong and Nelson \(2006\)](#).
- It can be used when the discrete feasible set is finite (i.e., fully constrained) or infinite (i.e., partially constrained or unconstrained).

- COMPASS for DOvS [Hong and Nelson \(2006\)](#):
  - Initialization:
  
  
  
  
  
  
  
  
  
  
  - At Iteration  $k$ :
    - Sampling:
  
  
  
  
  
  
  
  
  
  
    - Evaluation:
  
  
  
  
  
  
  
  
  
  
    - Updating:

- COMPASS for DOvS [Hong and Nelson \(2006\)](#):
  - **Initialization:** Arbitrarily choose  $\mathbf{x}_0 \in \mathcal{X}$ ; set  $\mathbf{x}_0^* = \mathbf{x}_0$  and  $\mathcal{V}_0 = \{\mathbf{x}_0\}$ ; take observations according to a simulation allocation rule (SAR) from  $\mathbf{x}_0$ ; let  $\mathcal{P}_0 = \mathcal{X}$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:**
    - **Evaluation:**
    - **Updating:**

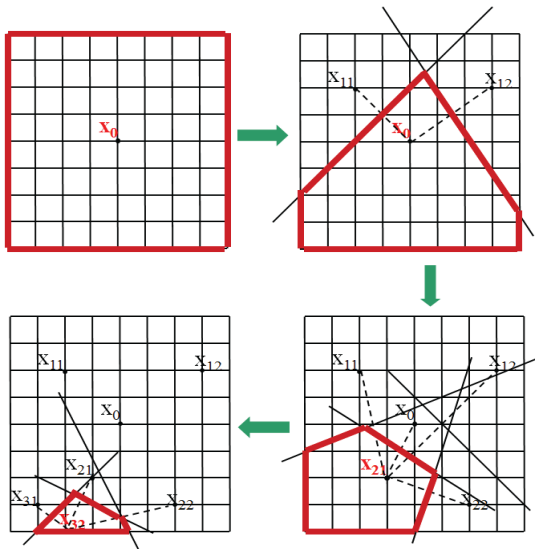
- COMPASS for DOvS [Hong and Nelson \(2006\)](#):
  - **Initialization:** Arbitrarily choose  $\mathbf{x}_0 \in \mathcal{X}$ ; set  $\mathbf{x}_0^* = \mathbf{x}_0$  and  $\mathcal{V}_0 = \{\mathbf{x}_0\}$ ; take observations according to a simulation allocation rule (SAR) from  $\mathbf{x}_0$ ; let  $\mathcal{P}_0 = \mathcal{X}$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:** Sample  $m$  solutions uniformly and independently from  $\mathcal{P}_k$ , denoted as  $\{\mathbf{x}_{k1}, \dots, \mathbf{x}_{km}\}$ ; let  $\mathcal{V}_{k+1} := \mathcal{V}_k \cup \{\mathbf{x}_{k1}, \dots, \mathbf{x}_{km}\}$  be the estimation set.
    - **Evaluation:**
    - **Updating:**

- COMPASS for DOvS [Hong and Nelson \(2006\)](#):
  - **Initialization:** Arbitrarily choose  $\mathbf{x}_0 \in \mathcal{X}$ ; set  $\mathbf{x}_0^* = \mathbf{x}_0$  and  $\mathcal{V}_0 = \{\mathbf{x}_0\}$ ; take observations according to a simulation allocation rule (SAR) from  $\mathbf{x}_0$ ; let  $\mathcal{P}_0 = \mathcal{X}$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:** Sample  $m$  solutions uniformly and independently from  $\mathcal{P}_k$ , denoted as  $\{\mathbf{x}_{k1}, \dots, \mathbf{x}_{km}\}$ ; let  $\mathcal{V}_{k+1} := \mathcal{V}_k \cup \{\mathbf{x}_{k1}, \dots, \mathbf{x}_{km}\}$  be the estimation set.
    - **Evaluation:** For each  $\mathbf{x} \in \mathcal{V}_{k+1}$ , take *additional* observations according to the SAR.
    - **Updating:**



- COMPASS for DOvS [Hong and Nelson \(2006\)](#):
  - **Initialization:** Arbitrarily choose  $\mathbf{x}_0 \in \mathcal{X}$ ; set  $\mathbf{x}_0^* = \mathbf{x}_0$  and  $\mathcal{V}_0 = \{\mathbf{x}_0\}$ ; take observations according to a simulation allocation rule (SAR) from  $\mathbf{x}_0$ ; let  $\mathcal{P}_0 = \mathcal{X}$ ; set iteration index  $k = 0$ .
  - **At Iteration  $k$ :**
    - **Sampling:** Sample  $m$  solutions uniformly and independently from  $\mathcal{P}_k$ , denoted as  $\{\mathbf{x}_{k1}, \dots, \mathbf{x}_{km}\}$ ; let  $\mathcal{V}_{k+1} := \mathcal{V}_k \cup \{\mathbf{x}_{k1}, \dots, \mathbf{x}_{km}\}$  be the estimation set.
    - **Evaluation:** For each  $\mathbf{x} \in \mathcal{V}_{k+1}$ , take *additional* observations according to the SAR.
    - **Updating:** Update  $\mathcal{P}_{k+1}$ ; choose the solution in  $\mathcal{V}_{k+1}$  with smallest estimated function value as  $\mathbf{x}_{k+1}^*$ ; set  $k \leftarrow k + 1$ .

- The way to construct  $\mathcal{P}_k$  – the most promising area:



- 1 Introduction
  - ▶ Definition
  - ▶ Types
- 2 White-box OvS Problem
  - ▶ Sample Average Approximation
- 3 Black-box COvS Problem
  - ▶ Gradient Descent
  - ▶ Stochastic Approximation
- 4 Black-box DOvS Problem
  - ▶ Simulated Annealing
  - ▶ COMPASS
- 5 Usage in Softwares

- In many commercial simulation softwares, like Arena, AnyLogic, Simio and FlexSim, OptQuest is integrated for simulation optimization.

- In many commercial simulation softwares, like Arena, AnyLogic, Simio and FlexSim, OptQuest is integrated for simulation optimization.
- OptQuest is based on a combination of methods, including linear/integer programming, heuristics and metaheuristics.
  - It is robust when used to solve practical OvS problems;
  - but it has no provable convergence for OvS problems.

- In many commercial simulation softwares, like Arena, AnyLogic, Simio and FlexSim, OptQuest is integrated for simulation optimization.
- OptQuest is based on a combination of methods, including linear/integer programming, heuristics and metaheuristics.
  - It is robust when used to solve practical OvS problems;
  - but it has no provable convergence for OvS problems.
- None of those OvS algorithms have been integrated into the commercial simulation softwares yet.

- In many commercial simulation softwares, like Arena, AnyLogic, Simio and FlexSim, OptQuest is integrated for simulation optimization.
- OptQuest is based on a combination of methods, including linear/integer programming, heuristics and metaheuristics.
  - It is robust when used to solve practical OvS problems;
  - but it has no provable convergence for OvS problems.
- None of those OvS algorithms have been integrated into the commercial simulation softwares yet.
- So, for researchers in the field of OvS, there is still a long way to go...