Balancing Exploitation and Exploration in Discrete Optimization via Simulation Through a Gaussian Process-Based Search

Lihua Sun, L. Jeff Hong, Zhaolin Hu

Presented by Qingkai Zhang

School of Management, Fudan University

2021.11.29

Outline

Outline

Random search algorithms

- To solve discrete optimization-via-simulation(DOvS) problems

$$\max_{\mathbf{x}\in\Theta} g(\mathbf{x}) := \mathrm{E}[G(\mathbf{x})]$$

  where the random variable $G(\mathbf{x})$ typically has no closed-form expression, and the solution set $\Theta = \Omega \cap \mathscr{Z}^d$.
- Locally convergent vs. Globally convergent algorithms
- Exploitation and Exploration trade-off

Random search algorithms

The random search framework:

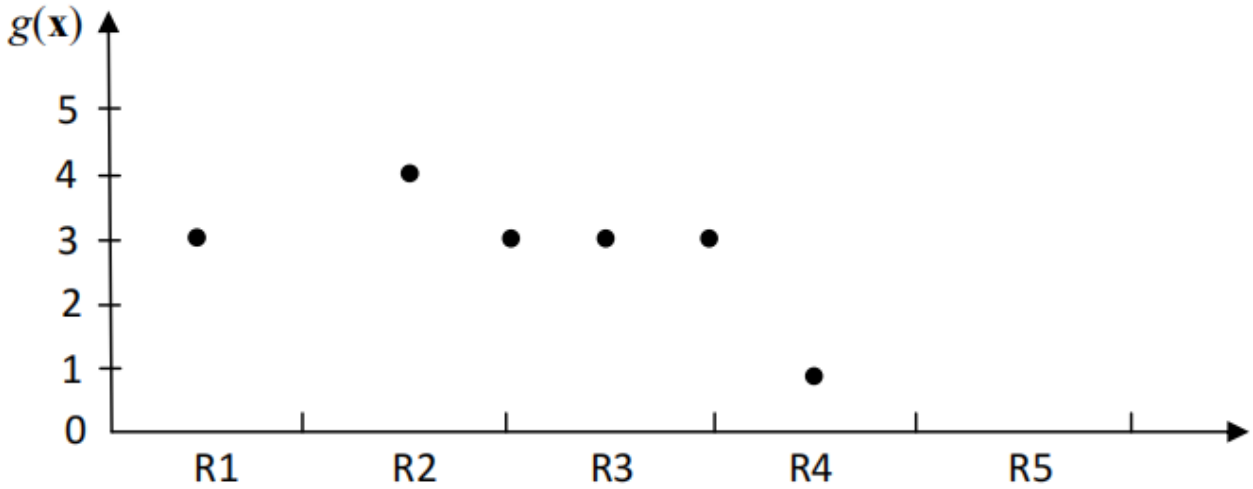
- Given $x_0 \in \Theta$ and let $k = 1$
- At iteration k :
  - Sampling: Determine a sampling distribution over Θ, denoted as $f_k(x \mid \mathcal{F}_{k-1})$. Sample a set of solutions based on $f_k(\cdot)$.
  - Evaluation: Evaluate (through running simulation experiments) the solutions and determine the current best solution x_k.
  - Let $k = k + 1$.

This paper derives a sampling distribution from a fast fitted Gaussian process, which has the desired properties and can automatically balance the exploitation and exploration trade-off.

Outline

Desired Properties of a Sampling Distribution

Consider a one-dimensional problem $\max g(x)$, where $g(x)$ can be evaluated without noise. Six solutions have been sampled and evaluated. Which region should have higher sampling probability?

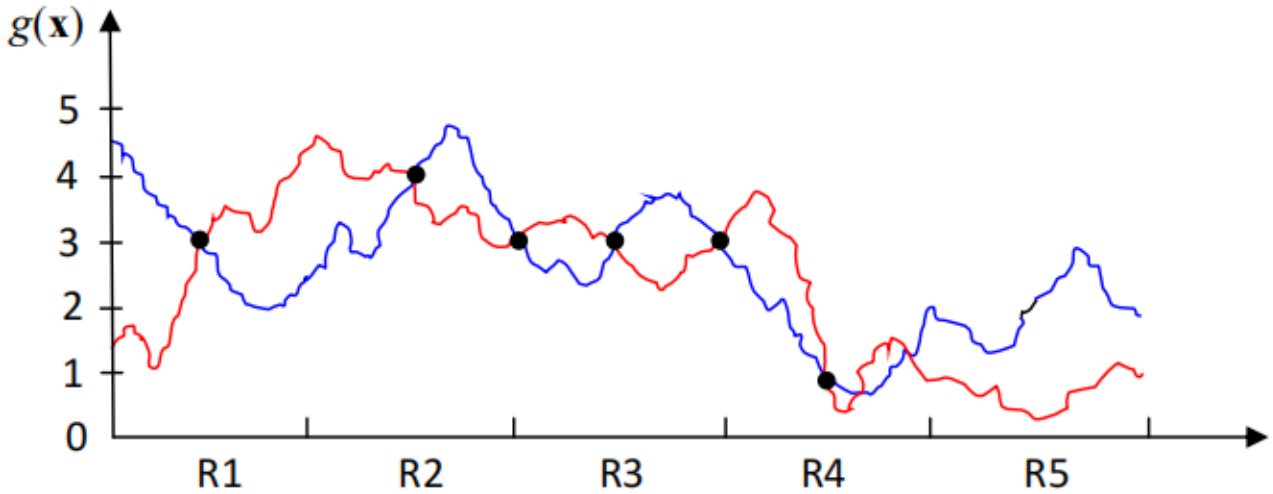


$R2$ vs. $R4$ $R1$ vs. $R3$ $R4$ vs. $R5$

It is the classical tradeoff between exploitation and exploration!

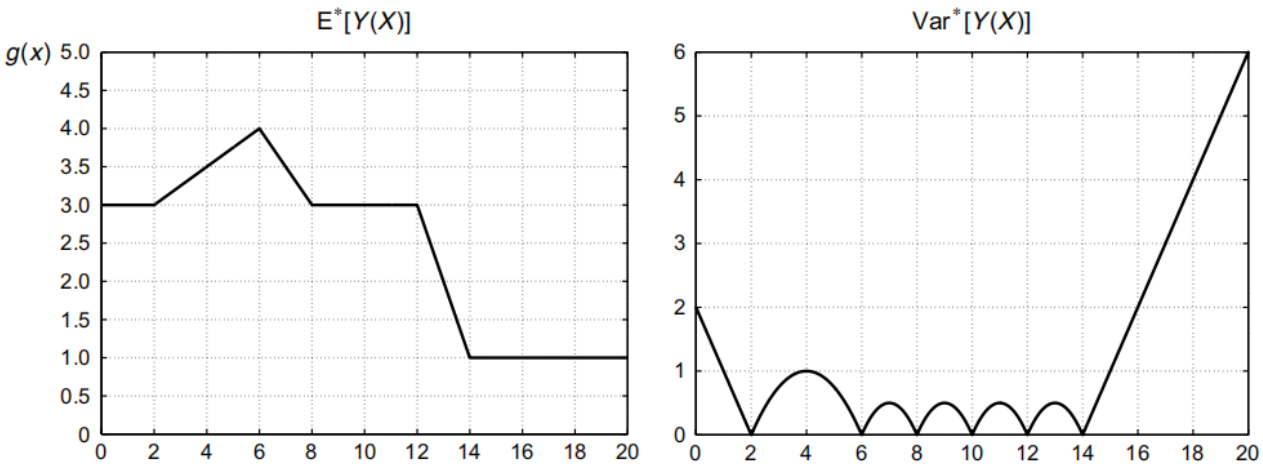
The Idea of Building a Sampling Distribution

- Suppose that $g(x)$ is a sample path of a Brownian motion $Y(x)$ process started from time $-\infty$.
- The process passes through the six points that have been evaluated.



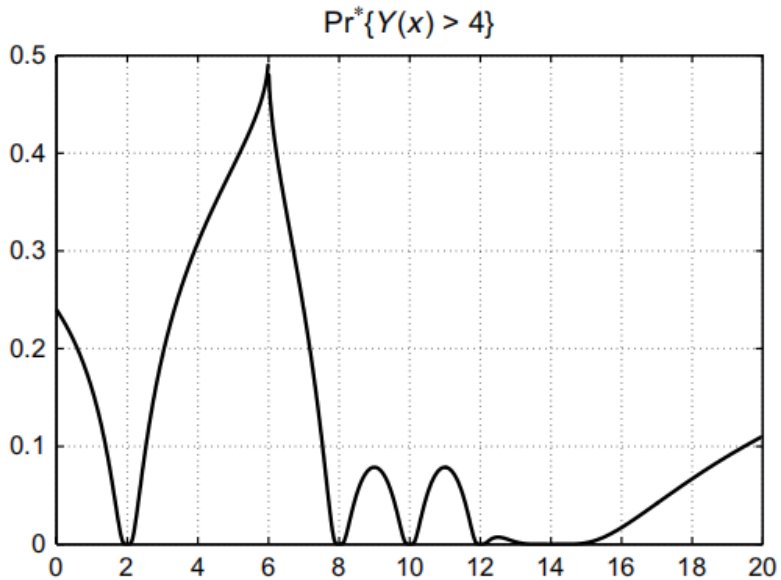
The Idea of Building a Sampling Distribution

- Given that the process passes through the six points, we can derive the condition distribution (i.e., condition mean and variance) of $Y(x)$ for any point $x_0 \in \Theta$



The Idea of Building a Sampling Distribution

- Given that the distributions, we can calculate $\Pr\left\{Y(x) > g^*_{k-1}\right\}$ for each $x \in \Theta$, where $g^*_{k-1} = 4$.



- Then, we normalize the probabilities into a sampling distribution, i.e.,

$$f_k(x) = \frac{\Pr\left\{Y(x) > g^*_{k-1}\right\}}{\sum_{y\in\Theta} \Pr\left\{Y(y) > g^*_{k-1}\right\}} \quad \forall x \in \Theta$$

The Idea of Building a Sampling Distribution

To apply this idea to solve DOvS problems, we have to study the following three issues:

- How to handle multi-dimensional problems?
- How to handle estimation errors?
- How to sample from the sampling distribution?

Outline

Kriging-Based Search

- The traditional kriging method models the simulation output at a point $\mathbf{x}$ as

$$G(\mathbf{x}) = M(\mathbf{x}) + \epsilon(\mathbf{x})$$

where $M(\mathbf{x})$ is a stationary Gaussian process with mean 0 and covariance function $\sigma^2\gamma(\cdot,\cdot)$, and $\epsilon(\mathbf{x})$ is a normal random variable with mean 0 and variance $\sigma_\epsilon^2(\mathbf{x})$.

- A stationary Gaussian process is a stochastic process $\left\{M(\mathbf{x}), \mathbf{x} \in \Re^d\right\}$ such that $M(\mathbf{x}) \sim N\left(0, \sigma^2\right)$ for any $\mathbf{x} \in \Re^d$ and $M\left(\mathbf{x_1}\right), \ldots, M\left(\mathbf{x_n}\right)$ are jointly normally distributed for any finite set of $\mathbf{x_1}, \ldots, \mathbf{x_n} \in \Re^d$
- The stationary Gaussian process $M(\mathbf{x})$ models the unknown objective function $g(\mathbf{x})$, the error term $\epsilon(\mathbf{x})$ models the noise in the simulation output,they are independent of each other. And we have $\mathrm{Cov}\left(\epsilon(\mathbf{x}), \epsilon\left(\mathbf{x}'\right)\right) = 0$ for any $\mathbf{x} \neq \mathbf{x}'$ in this paper.

Kriging-Based Search

- The traditional kriging method models the simulation output at a point $\mathbf{x}$ as

$$G(\mathbf{x}) = M(\mathbf{x}) + \epsilon(\mathbf{x})$$

  where $M(\mathbf{x})$ is a stationary Gaussian process with mean 0 and covariance function $\sigma^2\gamma(\cdot,\cdot)$, and $\epsilon(\mathbf{x})$ is a normal random variable with mean 0 and variance $\sigma_\epsilon^2(\mathbf{x})$.

- A stationary Gaussian process is a stochastic process $\left\{M(\mathbf{x}), \mathbf{x} \in \Re^d\right\}$ such that $M(\mathbf{x}) \sim N\left(0, \sigma^2\right)$ for any $\mathbf{x} \in \Re^d$ and $M\left(\mathbf{x_1}\right), \ldots, M\left(\mathbf{x_n}\right)$ are jointly normally distributed for any finite set of $\mathbf{x_1}, \ldots, \mathbf{x_n} \in \Re^d$

- The stationary Gaussian process $M(\mathbf{x})$ models the unknown objective function $g(\mathbf{x})$, the error term $\epsilon(\mathbf{x})$ models the noise in the simulation output,they are independent of each other. And we have $\mathrm{Cov}\left(\epsilon(\mathbf{x}), \epsilon\left(\mathbf{x}'\right)\right) = 0$ for any $\mathbf{x} \neq \mathbf{x}'$ in this paper.

Kriging-Based Search

- The traditional kriging method models the simulation output at a point $\mathbf{x}$ as

$$G(\mathbf{x}) = M(\mathbf{x}) + \epsilon(\mathbf{x})$$

  where $M(\mathbf{x})$ is a stationary Gaussian process with mean 0 and covariance function $\sigma^2\gamma(\cdot,\cdot)$, and $\epsilon(\mathbf{x})$ is a normal random variable with mean 0 and variance $\sigma_\epsilon^2(\mathbf{x})$.

- A stationary Gaussian process is a stochastic process $\left\{M(\mathbf{x}), \mathbf{x} \in \Re^d\right\}$ such that $M(\mathbf{x}) \sim N\left(0, \sigma^2\right)$ for any $\mathbf{x} \in \Re^d$ and $M\left(\mathbf{x_1}\right), \ldots, M\left(\mathbf{x_n}\right)$ are jointly normally distributed for any finite set of $\mathbf{x_1}, \ldots, \mathbf{x_n} \in \Re^d$

- The stationary Gaussian process $M(\mathbf{x})$ models the unknown objective function $g(\mathbf{x})$, the error term $\epsilon(\mathbf{x})$ models the noise in the simulation output,they are independent of each other. And we have $\mathrm{Cov}\left(\epsilon(\mathbf{x}), \epsilon\left(\mathbf{x}'\right)\right) = 0$ for any $\mathbf{x} \neq \mathbf{x}'$ in this paper.

Kriging-Based Search

- To model the spatial dependence, a correlation function

$$\gamma(\mathbf{x}_1, \mathbf{x}_2) = \text{Corr}(M(\mathbf{x}_1), M(\mathbf{x}_2))$$

  needs to be chosen, which is typically a function of $\|\mathbf{x}_1 - \mathbf{x}_2\|$.
- Different functions may lead to different level of smoothness of the sample path.
- In this paper, we use $\gamma(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\|\mathbf{x}_1 - \mathbf{x}_2\|^2\right)$.

Kriging-Based Search

- Suppose that, through the current iteration, a random search algorithm has visited m points, denoted as $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and have taken n_i simulation replications for $\mathbf{x}_i, i = 1, \ldots, m$.
- Let $\bar{G}(\mathbf{x}_i)$ denote the sample mean calculated from n_i observations of $G(\mathbf{x}_i)$ for all $i = 1, \ldots, m$, let $\bar{G} = \left(\bar{G}(\mathbf{x}_1), \ldots, \bar{G}(\mathbf{x}_m)\right)^{\mathrm{T}}$,
- and let Σ_ϵ be an $m \times m$ matrix whose (i, i) th element is $\sigma_\epsilon^2(\mathbf{x}_i)/n_i$ whereas all other elements are 0.
- let Γ be an $m \times m$ matrix whose (i, j) th element is $\gamma(\mathbf{x}_i, \mathbf{x}_j)$, and $\gamma(\mathbf{x}_0)$ be an m-dimensional vector whose i th element is $\gamma(\mathbf{x}_0, \mathbf{x}_i)$.

Kriging-Based Search

- Suppose that, through the current iteration, a random search algorithm has visited m points, denoted as $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and have taken n_i simulation replications for $\mathbf{x}_i, i = 1, \ldots, m$.
- Let $\bar{G}(\mathbf{x}_i)$ denote the sample mean calculated from n_i observations of $G(\mathbf{x}_i)$ for all $i = 1, \ldots, m$, let $\bar{G} = \left(\bar{G}(\mathbf{x}_1), \ldots, \bar{G}(\mathbf{x}_m)\right)^{\mathrm{T}}$,
- and let Σ_ϵ be an $m \times m$ matrix whose (i, i) th element is $\sigma_\epsilon^2(\mathbf{x}_i)/n_i$ whereas all other elements are 0.
- let Γ be an $m \times m$ matrix whose (i, j) th element is $\gamma(\mathbf{x}_i, \mathbf{x}_j)$, and $\gamma(\mathbf{x}_0)$ be an m-dimensional vector whose i th element is $\gamma(\mathbf{x}_0, \mathbf{x}_i)$.

Kriging-Based Search

- Suppose that, through the current iteration, a random search algorithm has visited m points, denoted as $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and have taken n_i simulation replications for $\mathbf{x}_i, i = 1, \ldots, m$.
- Let $\bar{G}(\mathbf{x}_i)$ denote the sample mean calculated from n_i observations of $G(\mathbf{x}_i)$ for all $i = 1, \ldots, m$, let $\bar{G} = \left(\bar{G}(\mathbf{x}_1), \ldots, \bar{G}(\mathbf{x}_m)\right)^{\mathrm{T}}$,
- and let Σ_ϵ be an $m \times m$ matrix whose (i, i) th element is $\sigma_\epsilon^2(\mathbf{x}_i)/n_i$ whereas all other elements are 0.
- let Γ be an $m \times m$ matrix whose (i, j) th element is $\gamma(\mathbf{x}_i, \mathbf{x}_j)$, and $\gamma(\mathbf{x}_0)$ be an m-dimensional vector whose i th element is $\gamma(\mathbf{x}_0, \mathbf{x}_i)$.

Kriging-Based Search

- Suppose that, through the current iteration, a random search algorithm has visited m points, denoted as $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and have taken n_i simulation replications for $\mathbf{x}_i, i = 1, \ldots, m$.
- Let $\bar{G}(\mathbf{x}_i)$ denote the sample mean calculated from n_i observations of $G(\mathbf{x}_i)$ for all $i = 1, \ldots, m$, let $\bar{G} = \left(\bar{G}(\mathbf{x}_1), \ldots, \bar{G}(\mathbf{x}_m)\right)^{\mathrm{T}}$,
- and let Σ_ϵ be an $m \times m$ matrix whose (i, i) th element is $\sigma_\epsilon^2(\mathbf{x}_i)/n_i$ whereas all other elements are 0.
- let Γ be an $m \times m$ matrix whose (i, j) th element is $\gamma(\mathbf{x}_i, \mathbf{x}_j)$, and $\gamma(\mathbf{x}_0)$ be an m-dimensional vector whose i th element is $\gamma(\mathbf{x}_0, \mathbf{x}_i)$.

Kriging-Based Search

- Then,[Ankenman et al. 2010] show that for any $\mathbf{x}_0$,
- the MSE-optimal linear predictor (stochastic kriging model) of $g(\mathbf{x}_0)$ conditioned on the observed data is

$$\hat{g}(\mathbf{x}_0) = \lambda(\mathbf{x}_0)^{\mathrm{T}} \bar{G}$$

with

$$\lambda(\mathbf{x}_0)^{\mathrm{T}} = \gamma(\mathbf{x}_0)^{\mathrm{T}} \left(\Gamma + \frac{1}{\sigma^2} \Sigma_\epsilon \right)^{-1}$$

- and the corresponding optimal MSE is

$$\mathrm{MSE}(\mathbf{x}_0) = \sigma^2 \left[1 - \lambda(\mathbf{x}_0)^{\mathrm{T}} \gamma(\mathbf{x}_0) \right]$$

Kriging-Based Search

- For every $\mathbf{x} \in \Theta$, let $Y'(\mathbf{x})$ denote the metamodel of $g(\mathbf{x})$ in the stochastic kriging model.
- Conditioned on all available information, $Y'(\mathbf{x})(= M(\mathbf{x}))$ is normally distributed with $\mathrm{E}^*[Y'(\mathbf{x})] = \hat{g}(\mathbf{x})$ and $\mathrm{Var}^*[Y'(\mathbf{x})] = \mathrm{MSE}(\mathbf{x})$,
- where we use the notation $\mathrm{E}^*(\cdot), \mathrm{Var}^*(\cdot)$ and $\Pr^*(\cdot)$ to denote they are conditioned on all available information.
- the mean function $\hat{g}(\mathbf{x})$ measures the need for exploitation and the variance function MSE $(\mathbf{x})$ measures the need for exploration.

Kriging-Based Search

- For every $\mathbf{x} \in \Theta$, let $Y'(\mathbf{x})$ denote the metamodel of $g(\mathbf{x})$ in the stochastic kriging model.
- Conditioned on all available information, $Y'(\mathbf{x})(= M(\mathbf{x}))$ is normally distributed with $\mathrm{E}^*[Y'(\mathbf{x})] = \hat{g}(\mathbf{x})$ and $\mathrm{Var}^*[Y'(\mathbf{x})] = \mathrm{MSE}(\mathbf{x})$,
- where we use the notation $\mathrm{E}^*(\cdot), \mathrm{Var}^*(\cdot)$ and $\Pr^*(\cdot)$ to denote they are conditioned on all available information.
- the mean function $\hat{g}(\mathbf{x})$ measures the need for exploitation and the variance function MSE $(\mathbf{x})$ measures the need for exploration.

Kriging-Based Search

- For every $\mathbf{x} \in \Theta$, let $Y'(\mathbf{x})$ denote the metamodel of $g(\mathbf{x})$ in the stochastic kriging model.
- Conditioned on all available information, $Y'(\mathbf{x})(= M(\mathbf{x}))$ is normally distributed with $\mathrm{E}^*[Y'(\mathbf{x})] = \hat{g}(\mathbf{x})$ and $\mathrm{Var}^*[Y'(\mathbf{x})] = \mathrm{MSE}(\mathbf{x})$,
- where we use the notation $\mathrm{E}^*(\cdot), \mathrm{Var}^*(\cdot)$ and $\mathrm{Pr}^*(\cdot)$ to denote they are conditioned on all available information.
- the mean function $\hat{g}(\mathbf{x})$ measures the need for exploitation and the variance function $\mathrm{MSE}(\mathbf{x})$ measures the need for exploration.

Kriging-Based Search

- For every $\mathbf{x} \in \Theta$, let $Y'(\mathbf{x})$ denote the metamodel of $g(\mathbf{x})$ in the stochastic kriging model.
- Conditioned on all available information, $Y'(\mathbf{x})(= M(\mathbf{x}))$ is normally distributed with $\mathrm{E}^*[Y'(\mathbf{x})] = \hat{g}(\mathbf{x})$ and $\mathrm{Var}^*[Y'(\mathbf{x})] = \mathrm{MSE}(\mathbf{x})$,
- where we use the notation $\mathrm{E}^*(\cdot), \mathrm{Var}^*(\cdot)$ and $\Pr^*(\cdot)$ to denote they are conditioned on all available information.
- the mean function $\hat{g}(\mathbf{x})$ measures the need for exploitation and the variance function MSE $(\mathbf{x})$ measures the need for exploration.

Kriging-Based Search

- Let c denote the current sample-best value, i.e., $c = \max\left\{\bar{G}(x_1), \ldots, \bar{G}(x_m)\right\}$.
- Then, the conditional probability $\Pr^*\{Y'(\mathbf{x}) > c\}$ may be calculated, and we may define the sampling distribution as

$$f(\mathbf{x}) = \frac{\Pr^*\{Y'(\mathbf{x}) > c\}}{\sum_{\mathbf{z}\in\Theta} \Pr^*\{Y'(\mathbf{z}) > c\}}, \quad \mathbf{x} \in \Theta$$

- For any $\mathbf{x} \in \Theta$, $\Pr^*\{Y'(\mathbf{x}) > c\}$ represents the conditional probability that $\mathbf{x}$ has a value that is better than the current sample-best value c
- $f(\mathbf{x})$ represents the relative importance of $\mathbf{x}$ among all solutions in Θ in its probability of being a better solution.
- $f(\mathbf{x})$ automatically balances the exploitation and exploration trade-off.

Kriging-Based Search

- Let c denote the current sample-best value, i.e., $c = \max\left\{\bar{G}(x_1), \ldots, \bar{G}(x_m)\right\}$.
- Then, the conditional probability $\Pr^*\{Y'(\mathbf{x}) > c\}$ may be calculated, and we may define the sampling distribution as

$$f(\mathbf{x}) = \frac{\Pr^*\{Y'(\mathbf{x}) > c\}}{\sum_{\mathbf{z}\in\Theta} \Pr^*\{Y'(\mathbf{z}) > c\}}, \quad \mathbf{x} \in \Theta$$

- For any $\mathbf{x} \in \Theta$, $\Pr^*\{Y'(\mathbf{x}) > c\}$ represents the conditional probability that $\mathbf{x}$ has a value that is better than the current sample-best value c
- $f(\mathbf{x})$ represents the relative importance of $\mathbf{x}$ among all solutions in Θ in its probability of being a better solution.
- $f(\mathbf{x})$ automatically balances the exploitation and exploration trade-off.

Kriging-Based Search

- Let c denote the current sample-best value, i.e., $c = \max\left\{\bar{G}(x_1), \ldots, \bar{G}(x_m)\right\}$.
- Then, the conditional probability $\Pr^*\{Y'(\mathbf{x}) > c\}$ may be calculated, and we may define the sampling distribution as

$$f(\mathbf{x}) = \frac{\Pr^*\{Y'(\mathbf{x}) > c\}}{\sum_{\mathbf{z}\in\Theta} \Pr^*\{Y'(\mathbf{z}) > c\}}, \quad \mathbf{x} \in \Theta$$

- For any $\mathbf{x} \in \Theta$, $\Pr^*\{Y'(\mathbf{x}) > c\}$ represents the conditional probability that $\mathbf{x}$ has a value that is better than the current sample-best value c
- $f(\mathbf{x})$ represents the relative importance of $\mathbf{x}$ among all solutions in Θ in its probability of being a better solution.
- $f(\mathbf{x})$ automatically balances the exploitation and exploration trade-off.

Kriging-Based Search

- Let c denote the current sample-best value, i.e., $c = \max\left\{\bar{G}(x_1), \ldots, \bar{G}(x_m)\right\}$.
- Then, the conditional probability $\Pr^*\{Y'(\mathbf{x}) > c\}$ may be calculated, and we may define the sampling distribution as

$$f(\mathbf{x}) = \frac{\Pr^*\{Y'(\mathbf{x}) > c\}}{\sum_{\mathbf{z}\in\Theta} \Pr^*\{Y'(\mathbf{z}) > c\}}, \quad \mathbf{x} \in \Theta$$

- For any $\mathbf{x} \in \Theta$, $\Pr^*\{Y'(\mathbf{x}) > c\}$ represents the conditional probability that $\mathbf{x}$ has a value that is better than the current sample-best value c
- $f(\mathbf{x})$ represents the relative importance of $\mathbf{x}$ among all solutions in Θ in its probability of being a better solution.
- $f(\mathbf{x})$ automatically balances the exploitation and exploration trade-off.

Kriging-Based Search

- Let c denote the current sample-best value, i.e., $c = \max\left\{\bar{G}(x_1), \ldots, \bar{G}(x_m)\right\}$.
- Then, the conditional probability $\Pr^*\{Y'(\mathbf{x}) > c\}$ may be calculated, and we may define the sampling distribution as

$$f(\mathbf{x}) = \frac{\Pr^*\{Y'(\mathbf{x}) > c\}}{\sum_{\mathbf{z}\in\Theta} \Pr^*\{Y'(\mathbf{z}) > c\}}, \quad \mathbf{x} \in \Theta$$

- For any $\mathbf{x} \in \Theta$, $\Pr^*\{Y'(\mathbf{x}) > c\}$ represents the conditional probability that $\mathbf{x}$ has a value that is better than the current sample-best value c
- $f(\mathbf{x})$ represents the relative importance of $\mathbf{x}$ among all solutions in Θ in its probability of being a better solution.
- $f(\mathbf{x})$ automatically balances the exploitation and exploration trade-off.

Fast Construction of Gaussian Process

- Note that in the stochastic kriging model, the prediction of the value at a given solution is determined through minimizing its MSE , which introduce the matrix inversion operation , it is computationally intensive when m is large.
- Because we need to fit a Gaussian process in each iteration, it is computationally NOT practical.
- It is important to notice that our goal is NOT to fit a surface, but to find a good sampling distribution. It is not necessary to minimize the MSE.

Fast Construction of Gaussian Process

- Note that in the stochastic kriging model, the prediction of the value at a given solution is determined through minimizing its MSE , which introduce the matrix inversion operation , it is computationally intensive when m is large.
- Because we need to fit a Gaussian process in each iteration, it is computationally NOT practical.
- It is important to notice that our goal is NOT to fit a surface, but to find a good sampling distribution. It is not necessary to minimize the MSE.

Fast Construction of Gaussian Process

- Note that in the stochastic kriging model, the prediction of the value at a given solution is determined through minimizing its MSE , which introduce the matrix inversion operation , it is computationally intensive when m is large.
- Because we need to fit a Gaussian process in each iteration, it is computationally NOT practical.
- It is important to notice that our goal is NOT to fit a surface, but to find a good sampling distribution. It is not necessary to minimize the MSE.

Fast Construction of Gaussian Process

We model $g(x)$ as a sample path of

$$Y(\mathbf{x}) = M(\mathbf{x}) + \lambda(\mathbf{x})'(\bar{G} - \mathbf{M}) + \lambda(\mathbf{x})'\epsilon$$

- $M(\mathbf{x})$ is a stationary Gaussian process,
- $\bar{G} = \left(\bar{G}(\mathbf{x}_1), \ldots, \bar{G}(\mathbf{x}_m)\right)^{\mathrm{T}}$ and $\mathbf{M} = (M(\mathbf{x}_1), \ldots, M(\mathbf{x}_m))'$
- $\epsilon = (\epsilon_1, \ldots, \epsilon_m)'$ where ϵ_i is an independent normal random variable with mean 0 and variance $\frac{\sigma_i^2}{n_i}, i = 1, \ldots, m$. The covariance matrix of ϵ is

$$\Sigma_\epsilon = \mathrm{diag}\left(\frac{\sigma_1^2}{n_1}, \ldots, \frac{\sigma_m^2}{n_m}\right)$$

Fast Construction of Gaussian Process

$\lambda(\mathbf{x})$ is a vector of weight functions that satisfy

Condition 2

For any $\mathbf{x} \in \Theta, \lambda(\mathbf{x})$ is continuous in $\mathbf{x}$ and satisfies

1. $\lambda_i(\mathbf{x}) \geqslant 0$ for any $i = 1, \ldots, m$;
2. $\sum_{i=1}^{m} \lambda_i(\mathbf{x}) = 1$;
3. $\lambda_i(\mathbf{x}_j) = 1\{\mathbf{x}_i = \mathbf{x}_j\}$ for all $i, j = 1, \ldots, m$, where $1\{\cdot\}$ is the indicator function.

For instance,for some $b > 0$

$$\lambda_i(\mathbf{x}) = \begin{cases} \frac{\|\mathbf{x}-\mathbf{x}_i\|^{-b}}{\sum_{j=1}^{m}\|\mathbf{x}-\mathbf{x}_j\|^{-b}} & \mathbf{x} \neq \mathbf{x}_i \\ 1 & \mathbf{x} = \mathbf{x}_i \end{cases}$$

Fast Construction of Gaussian Process

Proposition 1

For any $\mathbf{x} \in \Theta$, if Condition 2 is satisfied,

$$\begin{aligned}
\mathrm{E}^*[Y(\mathbf{x})] =& \lambda(\mathbf{x})^{\mathrm{T}} \bar{G}, \\
\mathrm{Var}^*[Y(\mathbf{x})] =& \sigma^2 \left[1 - 2\lambda(\mathbf{x})^{\mathrm{T}} \gamma(\mathbf{x}) + \lambda(\mathbf{x})^{\mathrm{T}} \Gamma \lambda(\mathbf{x})\right] \\
&+ \lambda(\mathbf{x})^{\mathrm{T}} \Sigma_\epsilon \lambda(\mathbf{x})
\end{aligned}$$

Furthermore,

$$\mathrm{E}^*\left[Y(\mathbf{x}_i)\right] = \bar{G}(\mathbf{x}_i)$$

$$\mathrm{Var}^*\left[Y(\mathbf{x}_i)\right] = \sigma^2(\mathbf{x}_i)/n_i$$

for all $i = 1, \ldots, m$

Fast Construction of Gaussian Process

- Proposition 1 provides an approach to calculating the $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$ functions without conducting matrix inversion.
- Proposition 1 also shows that the $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$ values at any evaluated solution $\mathbf{x}_i$ are its sample mean and the variance of the sample mean.
- Therefore, we rely solely on the simulation information to predict $g(\mathbf{x})$ if $\mathbf{x}$ has been simulated.
- Furthermore, when $n_i \to \infty$ or $\sigma^2(\mathbf{x}_i) = 0, \mathrm{E}^*[Y(\mathbf{x}_i)] \to g(\mathbf{x}_i)$ and $\mathrm{Var}^*[Y(\mathbf{x}_i)] \to 0$, for any $i = 1, \ldots, m$

Fast Construction of Gaussian Process

- Proposition 1 provides an approach to calculating the $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$ functions without conducting matrix inversion.
- Proposition 1 also shows that the $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$ values at any evaluated solution $\mathbf{x}_i$ are its sample mean and the variance of the sample mean.
- Therefore, we rely solely on the simulation information to predict $g(\mathbf{x})$ if $\mathbf{x}$ has been simulated.
- Furthermore, when $n_i \to \infty$ or $\sigma^2(\mathbf{x}_i) = 0, \mathrm{E}^*[Y(\mathbf{x}_i)] \to g(\mathbf{x}_i)$ and $\mathrm{Var}^*[Y(\mathbf{x}_i)] \to 0$, for any $i = 1, \ldots, m$

Fast Construction of Gaussian Process

- Proposition 1 provides an approach to calculating the $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$ functions without conducting matrix inversion.
- Proposition 1 also shows that the $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$ values at any evaluated solution $\mathbf{x}_i$ are its sample mean and the variance of the sample mean.
- Therefore, we rely solely on the simulation information to predict $g(\mathbf{x})$ if $\mathbf{x}$ has been simulated.
- Furthermore, when $n_i \to \infty$ or $\sigma^2(\mathbf{x}_i) = 0, \mathrm{E}^*[Y(\mathbf{x}_i)] \to g(\mathbf{x}_i)$ and $\mathrm{Var}^*[Y(\mathbf{x}_i)] \to 0$, for any $i = 1, \ldots, m$

Fast Construction of Gaussian Process

- Proposition 1 provides an approach to calculating the $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$ functions without conducting matrix inversion.
- Proposition 1 also shows that the $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$ values at any evaluated solution $\mathbf{x}_i$ are its sample mean and the variance of the sample mean.
- Therefore, we rely solely on the simulation information to predict $g(\mathbf{x})$ if $\mathbf{x}$ has been simulated.
- Furthermore, when $n_i \to \infty$ or $\sigma^2(\mathbf{x}_i) = 0, \mathrm{E}^*[Y(\mathbf{x}_i)] \to g(\mathbf{x}_i)$ and $\mathrm{Var}^*[Y(\mathbf{x}_i)] \to 0$, for any $i = 1, \ldots, m$

Fast Construction of Gaussian Process

- Proposition 1 shows that, for any solution $\mathbf{x}$ that has not been simulated, the $\mathrm{E}^*[Y(\mathbf{x})]$ is a linear combination of $\bar{G}(\mathbf{x}_i), i = 1, \ldots, m$, which are the sample means of the evaluated solutions.
- Notice that the commonly used weight function $\lambda(\mathbf{x})$, such as those introduced above, depends only on the distances of $\mathbf{x}$ to $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and closer solutions often have higher weights.
- Then, solutions that are close to the current sample-best point tend to have a high $\mathrm{E}^*[Y(\mathbf{x})]$ value and solutions that are close to the current sample-worst point tend to have a low $\mathrm{E}^*[Y(\mathbf{x})]$ value.
- Therefore, the $\mathrm{E}^*[Y(\mathbf{x})]$ function can successfully reflect the necessity of exploitation.

Fast Construction of Gaussian Process

- Proposition 1 shows that, for any solution $\mathbf{x}$ that has not been simulated, the $\mathrm{E}^*[Y(\mathbf{x})]$ is a linear combination of $\bar{G}(\mathbf{x}_i), i = 1, \ldots, m$, which are the sample means of the evaluated solutions.
- Notice that the commonly used weight function $\lambda(\mathbf{x})$, such as those introduced above, depends only on the distances of $\mathbf{x}$ to $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and closer solutions often have higher weights.
- Then, solutions that are close to the current sample-best point tend to have a high $\mathrm{E}^*[Y(\mathbf{x})]$ value and solutions that are close to the current sample-worst point tend to have a low $\mathrm{E}^*[Y(\mathbf{x})]$ value.
- Therefore, the $\mathrm{E}^*[Y(\mathbf{x})]$ function can successfully reflect the necessity of exploitation.

Fast Construction of Gaussian Process

- Proposition 1 shows that, for any solution $\mathbf{x}$ that has not been simulated, the $\mathrm{E}^*[Y(\mathbf{x})]$ is a linear combination of $\bar{G}(\mathbf{x}_i), i = 1, \ldots, m$, which are the sample means of the evaluated solutions.
- Notice that the commonly used weight function $\lambda(\mathbf{x})$, such as those introduced above, depends only on the distances of $\mathbf{x}$ to $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and closer solutions often have higher weights.
- Then, solutions that are close to the current sample-best point tend to have a high $\mathrm{E}^*[Y(\mathbf{x})]$ value and solutions that are close to the current sample-worst point tend to have a low $\mathrm{E}^*[Y(\mathbf{x})]$ value.
- Therefore, the $\mathrm{E}^*[Y(\mathbf{x})]$ function can successfully reflect the necessity of exploitation.

Fast Construction of Gaussian Process

- Proposition 1 shows that, for any solution $\mathbf{x}$ that has not been simulated, the $\mathrm{E}^*[Y(\mathbf{x})]$ is a linear combination of $\bar{G}(\mathbf{x}_i), i = 1, \ldots, m$, which are the sample means of the evaluated solutions.
- Notice that the commonly used weight function $\lambda(\mathbf{x})$, such as those introduced above, depends only on the distances of $\mathbf{x}$ to $\mathbf{x}_1, \ldots, \mathbf{x}_m$, and closer solutions often have higher weights.
- Then, solutions that are close to the current sample-best point tend to have a high $\mathrm{E}^*[Y(\mathbf{x})]$ value and solutions that are close to the current sample-worst point tend to have a low $\mathrm{E}^*[Y(\mathbf{x})]$ value.
- Therefore, the $\mathrm{E}^*[Y(\mathbf{x})]$ function can successfully reflect the necessity of exploitation.

Fast Construction of Gaussian Process

- Proposition 1 also shows that, for any solution $\mathbf{x}$ that has not been simulated, the $\mathrm{Var}^*[Y(\mathbf{x})]$ consists of two parts.
- The first part $\sigma^2\left[1-2\lambda(\mathbf{x})^{\mathrm{T}}\gamma(\mathbf{x})+\lambda(\mathbf{x})^{\mathrm{T}}\Gamma\lambda(\mathbf{x})\right]$ represents the uncertainty caused by the allocation of the simulated solutions (i.e., $\mathbf{x}_1,\ldots,\mathbf{x}_m$)
- and the second part $\lambda(\mathbf{x})^{\mathrm{T}}\Sigma_\epsilon\lambda(\mathbf{x})$ represents the uncertainty caused by the estimation error at the simulated solutions.
- To see how the $\mathrm{Var}^*[Y(\mathbf{x})]$ function reflects the necessity of exploration, we consider the first part of the function.

Fast Construction of Gaussian Process

- Proposition 1 also shows that, for any solution $\mathbf{x}$ that has not been simulated, the $\mathrm{Var}^*[Y(\mathbf{x})]$ consists of two parts.
- The first part $\sigma^2 \left[1 - 2\lambda(\mathbf{x})^{\mathrm{T}}\gamma(\mathbf{x}) + \lambda(\mathbf{x})^{\mathrm{T}}\Gamma\lambda(\mathbf{x})\right]$ represents the uncertainty caused by the allocation of the simulated solutions (i.e., $\mathbf{x}_1, \ldots, \mathbf{x}_m$)
- and the second part $\lambda(\mathbf{x})^{\mathrm{T}}\Sigma_\epsilon\lambda(\mathbf{x})$ represents the uncertainty caused by the estimation error at the simulated solutions.
- To see how the $\mathrm{Var}^*[Y(\mathbf{x})]$ function reflects the necessity of exploration, we consider the first part of the function.

Fast Construction of Gaussian Process

- Proposition 1 also shows that, for any solution $\mathbf{x}$ that has not been simulated, the $\mathrm{Var}^*[Y(\mathbf{x})]$ consists of two parts.
- The first part $\sigma^2 \left[1 - 2\lambda(\mathbf{x})^{\mathrm{T}}\gamma(\mathbf{x}) + \lambda(\mathbf{x})^{\mathrm{T}}\Gamma\lambda(\mathbf{x})\right]$ represents the uncertainty caused by the allocation of the simulated solutions (i.e., $\mathbf{x}_1, \ldots, \mathbf{x}_m$)
- and the second part $\lambda(\mathbf{x})^{\mathrm{T}}\Sigma_\epsilon\lambda(\mathbf{x})$ represents the uncertainty caused by the estimation error at the simulated solutions.
- To see how the $\mathrm{Var}^*[Y(\mathbf{x})]$ function reflects the necessity of exploration, we consider the first part of the function.

Fast Construction of Gaussian Process

- Proposition 1 also shows that, for any solution $\mathbf{x}$ that has not been simulated, the $\mathrm{Var}^*[Y(\mathbf{x})]$ consists of two parts.
- The first part $\sigma^2 \left[1 - 2\lambda(\mathbf{x})^{\mathrm{T}}\gamma(\mathbf{x}) + \lambda(\mathbf{x})^{\mathrm{T}}\Gamma\lambda(\mathbf{x})\right]$ represents the uncertainty caused by the allocation of the simulated solutions (i.e., $\mathbf{x}_1, \ldots, \mathbf{x}_m$)
- and the second part $\lambda(\mathbf{x})^{\mathrm{T}}\Sigma_\epsilon\lambda(\mathbf{x})$ represents the uncertainty caused by the estimation error at the simulated solutions.
- To see how the $\mathrm{Var}^*[Y(\mathbf{x})]$ function reflects the necessity of exploration, we consider the first part of the function.

Fast Construction of Gaussian Process

- We let

$$\tilde{\sigma}^2(\mathbf{x}) = \sigma^2 \left[1 - 2\lambda(\mathbf{x})^{\mathrm{T}}\gamma(\mathbf{x}) + \lambda(\mathbf{x})^{\mathrm{T}}\Gamma\lambda(\mathbf{x})\right]$$

  which is the first part of $\mathrm{Var}^*[Y(\mathbf{x})]$.
- A direct analysis of $\tilde{\sigma}^2(\mathbf{x})$ is difficult.However, we can analyze its lower bound.

Fast Construction of Gaussian Process

For the correlation function setting in the Gaussian process

$$\gamma(\mathbf{x}_1, \mathbf{x}_2) = h(\|\mathbf{x}_1 - \mathbf{x}_2\|)$$

We have the following condition and proposition

Condition 1

The correlation function $0 \leqslant h(t) \leqslant 1$ is a decreasing function of t when $t \geqslant 0$ and, for any
$\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2 \in \Theta, h(\|\mathbf{x}_1 - \mathbf{x}_2\|) \geqslant h(\|\mathbf{x}_0 - \mathbf{x}_1\|) \cdot h(\|\mathbf{x}_0 - \mathbf{x}_2\|)$

Proposition 2

Let $\underline{d}(\mathbf{x}) = \min\{\|\mathbf{x} - \mathbf{x}_1\|, \ldots, \|\mathbf{x} - \mathbf{x}_m\|\}$ Suppose that Condition 1 is satisfied by the correlation function $h(\cdot)$. Then,
$\tilde{\sigma}^2(\mathbf{x}) \geqslant \sigma^2[1 - h(\underline{d}(\mathbf{x}))]^2$ and $\tilde{\sigma}^2(\mathbf{x}) = 0$ if $\underline{d}(\mathbf{x}) = 0$

Fast Construction of Gaussian Process

- By Proposition 2, the lower bound of $\tilde{\sigma}^2(\mathbf{x})$ increases as $\mathbf{x}$ moves away from the set of evaluated solutions and decreases as $\mathbf{x}$ moves closer to one of the evaluated solutions.
- Therefore, we conclude that the $\mathrm{Var}^*[Y(\mathbf{x})]$ function reflects the necessity of exploration.
- From the above analysis we see the model has the appealing properties and is appropriate in deriving sampling distributions.
- Then, given the values of $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$, we may calculate the $\Pr^*\{Y(\mathbf{x}) > c\}$ where $c = \max\left\{\bar{G}(\mathbf{x}_1), \ldots, \bar{G}(\mathbf{x}_m)\right\}$ and derive the sampling distribution.

$$f(\mathbf{x}) = \frac{\Pr^*\{Y'(\mathbf{x}) > c\}}{\sum_{\mathbf{z}\in\Theta} \Pr^*\{Y'(\mathbf{z}) > c\}}, \quad \mathbf{x} \in \Theta$$

Fast Construction of Gaussian Process

- By Proposition 2, the lower bound of $\tilde{\sigma}^2(\mathbf{x})$ increases as $\mathbf{x}$ moves away from the set of evaluated solutions and decreases as $\mathbf{x}$ moves closer to one of the evaluated solutions.
- Therefore, we conclude that the $\mathrm{Var}^*[Y(\mathbf{x})]$ function reflects the necessity of exploration.
- From the above analysis we see the model has the appealing properties and is appropriate in deriving sampling distributions.
- Then, given the values of $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$, we may calculate the $\Pr^*\{Y(\mathbf{x}) > c\}$ where $c = \max\left\{\bar{G}(\mathbf{x}_1), \ldots, \bar{G}(\mathbf{x}_m)\right\}$ and derive the sampling distribution.

$$f(\mathbf{x}) = \frac{\Pr^*\{Y'(\mathbf{x}) > c\}}{\sum_{\mathbf{z}\in\Theta} \Pr^*\{Y'(\mathbf{z}) > c\}}, \quad \mathbf{x} \in \Theta$$

Fast Construction of Gaussian Process

- By Proposition 2, the lower bound of $\tilde{\sigma}^2(\mathbf{x})$ increases as $\mathbf{x}$ moves away from the set of evaluated solutions and decreases as $\mathbf{x}$ moves closer to one of the evaluated solutions.
- Therefore, we conclude that the $\mathrm{Var}^*[Y(\mathbf{x})]$ function reflects the necessity of exploration.
- From the above analysis we see the model has the appealing properties and is appropriate in deriving sampling distributions.
- Then, given the values of $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$, we may calculate the $\Pr^*\{Y(\mathbf{x}) > c\}$ where $c = \max\left\{\bar{G}(\mathbf{x}_1), \ldots, \bar{G}(\mathbf{x}_m)\right\}$ and derive the sampling distribution.

$$f(\mathbf{x}) = \frac{\Pr^*\{Y'(\mathbf{x}) > c\}}{\sum_{\mathbf{z}\in\Theta} \Pr^*\{Y'(\mathbf{z}) > c\}}, \quad \mathbf{x} \in \Theta$$

Fast Construction of Gaussian Process

- By Proposition 2, the lower bound of $\tilde{\sigma}^2(\mathbf{x})$ increases as $\mathbf{x}$ moves away from the set of evaluated solutions and decreases as $\mathbf{x}$ moves closer to one of the evaluated solutions.
- Therefore, we conclude that the $\mathrm{Var}^*[Y(\mathbf{x})]$ function reflects the necessity of exploration.
- From the above analysis we see the model has the appealing properties and is appropriate in deriving sampling distributions.
- Then, given the values of $\mathrm{E}^*[Y(\mathbf{x})]$ and $\mathrm{Var}^*[Y(\mathbf{x})]$, we may calculate the $\Pr^*\{Y(\mathbf{x}) > c\}$ where $c = \max\left\{\bar{G}(\mathbf{x}_1), \ldots, \bar{G}(\mathbf{x}_m)\right\}$ and derive the sampling distribution.

$$f(\mathbf{x}) = \frac{\Pr^*\{Y'(\mathbf{x}) > c\}}{\sum_{\mathbf{z}\in\Theta} \Pr^*\{Y'(\mathbf{z}) > c\}}, \quad \mathbf{x} \in \Theta$$

Sampling from the Sampling Distribution

- Note that

$$f_k(x) = \frac{\Pr\left\{Y(x) > \widehat{g}^*_{k-1}\right\}}{\sum_{y\in\Theta}\Pr\left\{Y(y) > \widehat{g}^*_{k-1}\right\}} \quad \forall x \in \Theta$$

  It is often impossible to compute $\sum_{y\in\Theta}\Pr\left\{Y(y) > \widehat{g}^*_{k-1}\right\}$.

- Note that $\Pr\left\{Y(x) > \hat{g}^*_{k-1}\right\} \leq \frac{1}{2}$ for all $x \in \Theta$. Then, we use the following Acceptance-Rejection Sampling (ARS) Algorithm to sample from $f_k(x)$:

  Step 1 Sample Z uniformly from Θ,

  Step 2 Sample U from $U(0,1)$. If $U \leq 2\Pr\left\{Y(Z) > \widehat{g}^*_{k-1}\right\}$, let $X = Z$; otherwise, go to Step 1,

  Step 3 Output X.

Sampling from the Sampling Distribution

- Note that

$$f_k(x) = \frac{\Pr\left\{Y(x) > \widehat{g}^*_{k-1}\right\}}{\sum_{y\in\Theta}\Pr\left\{Y(y) > \widehat{g}^*_{k-1}\right\}} \quad \forall x \in \Theta$$

  It is often impossible to compute $\sum_{y\in\Theta}\Pr\left\{Y(y) > \widehat{g}^*_{k-1}\right\}$.

- Note that $\Pr\left\{Y(x) > \widehat{g}^*_{k-1}\right\} \leq \frac{1}{2}$ for all $x \in \Theta$. Then, we use the following Acceptance-Rejection Sampling (ARS) Algorithm to sample from $f_k(x)$:

  Step 1 Sample Z uniformly from Θ,

  Step 2 Sample U from $U(0,1)$. If $U \leq 2\Pr\left\{Y(Z) > \widehat{g}^*_{k-1}\right\}$, let $X = Z$; otherwise, go to Step 1,

  Step 3 Output X.

Sampling from the Sampling Distribution

- Note that

$$f_k(x) = \frac{\Pr\left\{Y(x) > \widehat{g}^*_{k-1}\right\}}{\sum_{y\in\Theta}\Pr\left\{Y(y) > \widehat{g}^*_{k-1}\right\}} \quad \forall x \in \Theta$$

  It is often impossible to compute $\sum_{y\in\Theta}\Pr\left\{Y(y) > \widehat{g}^*_{k-1}\right\}$.

- Note that $\Pr\left\{Y(x) > \widehat{g}^*_{k-1}\right\} \leq \frac{1}{2}$ for all $x \in \Theta$. Then, we use the following Acceptance-Rejection Sampling (ARS) Algorithm to sample from $f_k(x)$:

  Step 1 Sample Z uniformly from Θ,

  Step 2 Sample U from $U(0,1)$. If $U \leq 2\Pr\left\{Y(Z) > \widehat{g}^*_{k-1}\right\}$, let $X = Z$; otherwise, go to Step 1,

  Step 3 Output X.

Sampling from the Sampling Distribution

- Note that

$$f_k(x) = \frac{\Pr\left\{Y(x) > \widehat{g}^*_{k-1}\right\}}{\sum_{y\in\Theta}\Pr\left\{Y(y) > \widehat{g}^*_{k-1}\right\}} \quad \forall x \in \Theta$$

  It is often impossible to compute $\sum_{y\in\Theta}\Pr\left\{Y(y) > \widehat{g}^*_{k-1}\right\}$.

- Note that $\Pr\left\{Y(x) > \hat{g}^*_{k-1}\right\} \leq \frac{1}{2}$ for all $x \in \Theta$. Then, we use the following Acceptance-Rejection Sampling (ARS) Algorithm to sample from $f_k(x)$:

  Step 1 Sample Z uniformly from Θ,

  Step 2 Sample U from $U(0,1)$. If $U \leq 2\Pr\left\{Y(Z) > \widehat{g}^*_{k-1}\right\}$, let $X = Z$; otherwise, go to Step 1,

  Step 3 Output X.

Sampling from the Sampling Distribution

- Note that

$$f_k(x) = \frac{\Pr\left\{Y(x) > \widehat{g}^*_{k-1}\right\}}{\sum_{y\in\Theta}\Pr\left\{Y(y) > \widehat{g}^*_{k-1}\right\}} \quad \forall x \in \Theta$$

  It is often impossible to compute $\sum_{y\in\Theta}\Pr\left\{Y(y) > \widehat{g}^*_{k-1}\right\}$.

- Note that $\Pr\left\{Y(x) > \widehat{g}^*_{k-1}\right\} \leq \frac{1}{2}$ for all $x \in \Theta$. Then, we use the following Acceptance-Rejection Sampling (ARS) Algorithm to sample from $f_k(x)$:

  Step 1 Sample Z uniformly from Θ,

  Step 2 Sample U from $U(0,1)$. If $U \leq 2\Pr\left\{Y(Z) > \widehat{g}^*_{k-1}\right\}$, let $X = Z$; otherwise, go to Step 1,

  Step 3 Output X.

Sampling from the Sampling Distribution

- By applying the ARS algorithm, we avoid computing the closed-form expression of $f(\mathbf{x})$ and, thus, significantly improve the efficiency of sampling from $f(\mathbf{x})$.
- When the probability mass of $f(\mathbf{x})$ is mainly concentrated on small subsets of Θ, however, the acceptance-rejection scheme used in the ARS algorithm may no longer be efficient because the probability of acceptance may be very low.
- Therefore, we also develop an approximate sampling algorithm by using a Markov chain sampling approach, which called Markov Chain Coordinate Sampling (MCCS)Algorithm.

Sampling from the Sampling Distribution

- By applying the ARS algorithm, we avoid computing the closed-form expression of $f(\mathbf{x})$ and, thus, significantly improve the efficiency of sampling from $f(\mathbf{x})$.
- When the probability mass of $f(\mathbf{x})$ is mainly concentrated on small subsets of Θ, however, the acceptance-rejection scheme used in the ARS algorithm may no longer be efficient because the probability of acceptance may be very low.
- Therefore, we also develop an approximate sampling algorithm by using a Markov chain sampling approach, which called Markov Chain Coordinate Sampling (MCCS)Algorithm.

Sampling from the Sampling Distribution

- By applying the ARS algorithm, we avoid computing the closed-form expression of $f(\mathbf{x})$ and, thus, significantly improve the efficiency of sampling from $f(\mathbf{x})$.
- When the probability mass of $f(\mathbf{x})$ is mainly concentrated on small subsets of Θ, however, the acceptance-rejection scheme used in the ARS algorithm may no longer be efficient because the probability of acceptance may be very low.
- Therefore, we also develop an approximate sampling algorithm by using a Markov chain sampling approach, which called Markov Chain Coordinate Sampling (MCCS)Algorithm.

Sampling from the Sampling Distribution

Step 0 Let $t = 0, \mathbf{y} = \mathbf{x}_0$.

Step 1 Let $t = t + 1$. Sample uniformly an integer l from 1 to d. Let $l(\mathbf{y}, l)$ be the line that passes through $\mathbf{y}$ and parallel to the $\mathbf{y}_l$ coordinate axis. Then $l(\mathbf{y}, l)$ intersects with the boundary of Ω (notice that $\Theta \subset \Omega$ and $\Omega \subset \Re^d$ is a convex set) at two points c_1, c_2. Sample an integer j uniformly from $[c_1, \mathbf{y}(l) - 1] \cup [\mathbf{y}(l) + 1, c_2]$. Set $\mathbf{z} = \mathbf{y}$ and then set $\mathbf{z}(l) = j$.

Step 2 If $U \leqslant f(\mathbf{z})/f(\mathbf{y}) = \Pr^*\{Y(\mathbf{z}) > c\}/\Pr^*\{Y(\mathbf{y}) > c\}$, set $\mathbf{y} = \mathbf{z}$.

Step 3 If $t = T$, return $\mathbf{y}$; otherwise go to Step 1 .

Sampling from the Sampling Distribution

- According to [Baumert et al. 2010], we have, as $T \to \infty$, the distribution of $\mathbf{y}$ converges to the sampling distribution $f(\cdot)$.
- And the MCCS algorithm guarantees to sample (approximately) a solution every T steps.
- Therefore, it may become more efficient than the ARS algorithm when the acceptance rate in the ARS becomes very low (i.e.,lower than $\frac{1}{T}$).
- Therefore, we may use the MCCS algorithm when the ARS algorithm becomes slow.

Outline

GPS Algorithm for DOvS Problems

- In the GPS algorithm, we let $\mathcal{S}_k$ denote the sets of simulated solutions through iteration k.

  Step 0 Sample ℓ solutions uniformly from Θ, denoted as $x_{01}, \ldots, x_{0\ell}$. Take m_0 observations for each of them and calculate their sample means and sample variances. Let $\mathcal{S}_0 = \{x_{01}, \ldots, x_{0\ell}\}$. Let $x_0^* = \text{argmax}_{x \in \mathcal{S}_0} \bar{G}(x)$ and $\widehat{g}_0^* = \bar{G}\left(x_0^*\right)$. Set $k = 0$

  Step 1 Let $k = k+1$. Construct a sampling distribution $f_k(x)$ based on all solutions in $\mathcal{S}_{k-1}$. Sample ℓ solutions from Θ based on $f_k(x)$, denoted as $x_{k1}, \ldots, x_{k\ell}$.

  Step 2 Take m_k observations for x_{k-1}^* and $x_{k1}, \ldots, x_{k\ell}$ and update their sample means and sample variances. Let $x_k^* = \text{argmax}_{x \in \mathcal{S}_k} \bar{G}(x)$ and $\widehat{g}_k^* = \bar{G}(x_k^*)$. Go to Step **1**.

Convergence of GPS Algorithm

Lemma 4

Let $n_k(\mathbf{x})$ denote the number of simulation observations through iteration k for all $\mathbf{x} \in \mathcal{S}_k$. Suppose that the GPS algorithm is used to solve Problem (1) and Conditions 1 and 2 are satisfied. Then, $n_k(\mathbf{x}) \to \infty$ w.p. 1 for all $\mathbf{x} \in \Theta$

Theorem 1

Suppose that the GPS algorithm is used to solve Problem (1) and Conditions 1 and 2 are satisfied. Then, $\hat{g}_k^* \to g^*$ w.p.1 as $k \to \infty$.

Stopping Criteria

$$\Delta_{k,1} = \frac{1}{|\Theta|} \sum_{\mathbf{x} \in \Theta} \Pr^* \left\{ Y_k(\mathbf{x}) \geqslant \hat{g}_k^* \right\}$$

$$\Delta_{k,2} = \frac{1}{|\Theta|} \sum_{\mathbf{x} \in \Theta} \mathrm{E}^* \left[\left(Y_k(\mathbf{x}) - \hat{g}_k^* \right)^+ \right]$$

$$\Delta_{k,3} = \sum_{\mathbf{x} \in \Theta} \Pr^* \left\{ Y_k(\mathbf{x}) \geqslant \hat{g}_k^* \right\} f_{k+1}(\mathbf{x})$$

$$\Delta_{k,4} = \sum_{\mathbf{x} \in \Theta} \mathrm{E}^* \left[\left(Y_k(\mathbf{x}) - \hat{g}_k^* \right)^+ \right] f_{k+1}(\mathbf{x})$$

Stopping Criteria

- Notice that $\Delta_{k,1} = \Pr^* \{ Y_k(\mathbf{U}) \geqslant \hat{g}_k^* \}$ and $\Delta_{k,2} = \mathrm{E}^* \left[(Y_k(\mathbf{U}) - \hat{g}_k^*)^+ \right]$, where $\mathbf{U}$ is a random vector that is uniformly distributed on Θ and independent of other randomness.
- $\Delta_{k,1}$ is the expected conditional probability that a uniformly generated solution on Θ has a value that is greater than the current sample best solution
- $\Delta_{k,2}$ is the expected conditional improvement.
- Therefore, we may stop the algorithm when the values of $\Delta_{k,1}$ and $\Delta_{k,2}$ are small enough,
- meaning that the chance of finding a better solution is small enough so that we may not have a significant loss if we stop the algorithm at iteration k.

Stopping Criteria

- Notice that $\Delta_{k,1} = \Pr^* \{ Y_k(\mathbf{U}) \geqslant \hat{g}_k^* \}$ and $\Delta_{k,2} = \mathrm{E}^* \left[(Y_k(\mathbf{U}) - \hat{g}_k^*)^+ \right]$, where $\mathbf{U}$ is a random vector that is uniformly distributed on Θ and independent of other randomness.
- $\Delta_{k,1}$ is the expected conditional probability that a uniformly generated solution on Θ has a value that is greater than the current sample best solution
- $\Delta_{k,2}$ is the expected conditional improvement.
- Therefore, we may stop the algorithm when the values of $\Delta_{k,1}$ and $\Delta_{k,2}$ are small enough,
- meaning that the chance of finding a better solution is small enough so that we may not have a significant loss if we stop the algorithm at iteration k.

Stopping Criteria

- Notice that $\Delta_{k,1} = \Pr^* \{ Y_k(\mathbf{U}) \geqslant \hat{g}_k^* \}$ and $\Delta_{k,2} = \mathrm{E}^* \left[(Y_k(\mathbf{U}) - \hat{g}_k^*)^+ \right]$, where $\mathbf{U}$ is a random vector that is uniformly distributed on Θ and independent of other randomness.
- $\Delta_{k,1}$ is the expected conditional probability that a uniformly generated solution on Θ has a value that is greater than the current sample best solution
- $\Delta_{k,2}$ is the expected conditional improvement.
- Therefore, we may stop the algorithm when the values of $\Delta_{k,1}$ and $\Delta_{k,2}$ are small enough,
- meaning that the chance of finding a better solution is small enough so that we may not have a significant loss if we stop the algorithm at iteration k.

Stopping Criteria

- Notice that $\Delta_{k,1} = \Pr^* \{ Y_k(\mathbf{U}) \geqslant \hat{g}_k^* \}$ and $\Delta_{k,2} = \mathrm{E}^* \left[(Y_k(\mathbf{U}) - \hat{g}_k^*)^+ \right]$, where $\mathbf{U}$ is a random vector that is uniformly distributed on Θ and independent of other randomness.
- $\Delta_{k,1}$ is the expected conditional probability that a uniformly generated solution on Θ has a value that is greater than the current sample best solution
- $\Delta_{k,2}$ is the expected conditional improvement.
- Therefore, we may stop the algorithm when the values of $\Delta_{k,1}$ and $\Delta_{k,2}$ are small enough,
- meaning that the chance of finding a better solution is small enough so that we may not have a significant loss if we stop the algorithm at iteration k.

Stopping Criteria

- Notice that $\Delta_{k,1} = \Pr^* \{ Y_k(\mathbf{U}) \geqslant \hat{g}_k^* \}$ and $\Delta_{k,2} = \mathrm{E}^* \left[(Y_k(\mathbf{U}) - \hat{g}_k^*)^+ \right]$, where $\mathbf{U}$ is a random vector that is uniformly distributed on Θ and independent of other randomness.
- $\Delta_{k,1}$ is the expected conditional probability that a uniformly generated solution on Θ has a value that is greater than the current sample best solution
- $\Delta_{k,2}$ is the expected conditional improvement.
- Therefore, we may stop the algorithm when the values of $\Delta_{k,1}$ and $\Delta_{k,2}$ are small enough,
- meaning that the chance of finding a better solution is small enough so that we may not have a significant loss if we stop the algorithm at iteration k.

Stopping Criteria

- Similarly, $\Delta_{k,3} = \Pr^* \{Y_k(\mathbf{X}) \geqslant \hat{g}_k^*\}$ and $\Delta_{k,4} = \mathrm{E}^* \left[(Y_k(\mathbf{X}) - \hat{g}_k^*)^+\right]$, where $\mathbf{X}$ is a random vector that is distributed according to $f_{k+1}(\mathbf{x})$ and independent of other randomness.
- $\Delta_{k,3}$ is the expected conditional probability that a better solution can be found in iteration $k+1$
- $\Delta_{k,4}$ is the expected conditional improvement in iteration $k+1$.
- we may stop the algorithm when the values of $\Delta_{k,3}$ and $\Delta_{k,4}$ are small enough,
- indicating that the gain from an additional iteration is small enough

Stopping Criteria

- Similarly, $\Delta_{k,3} = \Pr^* \{Y_k(\mathbf{X}) \geqslant \hat{g}_k^*\}$ and $\Delta_{k,4} = \mathrm{E}^* \left[(Y_k(\mathbf{X}) - \hat{g}_k^*)^+\right]$, where $\mathbf{X}$ is a random vector that is distributed according to $f_{k+1}(\mathbf{x})$ and independent of other randomness.
- $\Delta_{k,3}$ is the expected conditional probability that a better solution can be found in iteration $k+1$
- $\Delta_{k,4}$ is the expected conditional improvement in iteration $k+1$.
- we may stop the algorithm when the values of $\Delta_{k,3}$ and $\Delta_{k,4}$ are small enough,
- indicating that the gain from an additional iteration is small enough

Stopping Criteria

- Similarly, $\Delta_{k,3} = \Pr^* \{Y_k(\mathbf{X}) \geqslant \hat{g}_k^*\}$ and $\Delta_{k,4} = \mathrm{E}^* \left[(Y_k(\mathbf{X}) - \hat{g}_k^*)^+\right]$, where $\mathbf{X}$ is a random vector that is distributed according to $f_{k+1}(\mathbf{x})$ and independent of other randomness.
- $\Delta_{k,3}$ is the expected conditional probability that a better solution can be found in iteration $k+1$
- $\Delta_{k,4}$ is the expected conditional improvement in iteration $k+1$.
- we may stop the algorithm when the values of $\Delta_{k,3}$ and $\Delta_{k,4}$ are small enough,
- indicating that the gain from an additional iteration is small enough

Stopping Criteria

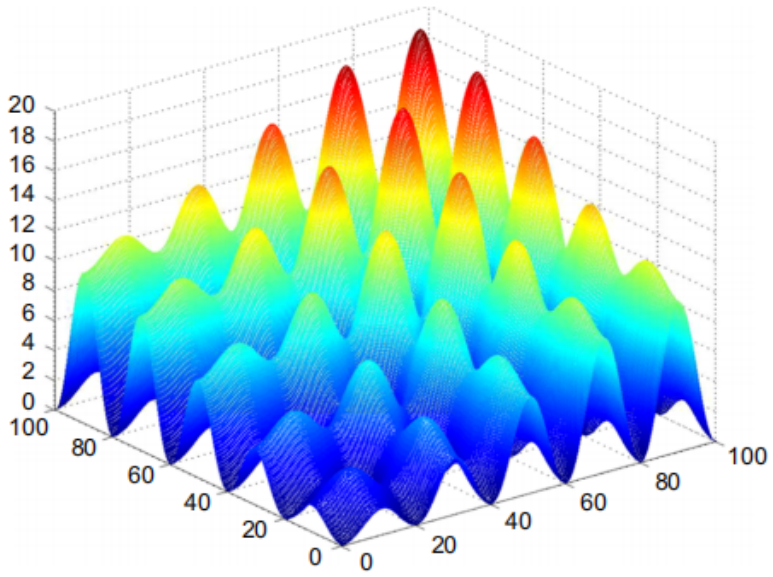
- Similarly, $\Delta_{k,3} = \Pr^* \{Y_k(\mathbf{X}) \geqslant \hat{g}_k^*\}$ and $\Delta_{k,4} = \mathrm{E}^* \left[(Y_k(\mathbf{X}) - \hat{g}_k^*)^+\right]$, where $\mathbf{X}$ is a random vector that is distributed according to $f_{k+1}(\mathbf{x})$ and independent of other randomness.
- $\Delta_{k,3}$ is the expected conditional probability that a better solution can be found in iteration $k+1$
- $\Delta_{k,4}$ is the expected conditional improvement in iteration $k+1$.
- we may stop the algorithm when the values of $\Delta_{k,3}$ and $\Delta_{k,4}$ are small enough,
- indicating that the gain from an additional iteration is small enough

Stopping Criteria

- Similarly, $\Delta_{k,3} = \Pr^* \{Y_k(\mathbf{X}) \geqslant \hat{g}_k^*\}$ and $\Delta_{k,4} = \mathrm{E}^* \left[(Y_k(\mathbf{X}) - \hat{g}_k^*)^+\right]$, where $\mathbf{X}$ is a random vector that is distributed according to $f_{k+1}(\mathbf{x})$ and independent of other randomness.
- $\Delta_{k,3}$ is the expected conditional probability that a better solution can be found in iteration $k+1$
- $\Delta_{k,4}$ is the expected conditional improvement in iteration $k+1$.
- we may stop the algorithm when the values of $\Delta_{k,3}$ and $\Delta_{k,4}$ are small enough,
- indicating that the gain from an additional iteration is small enough

Outline

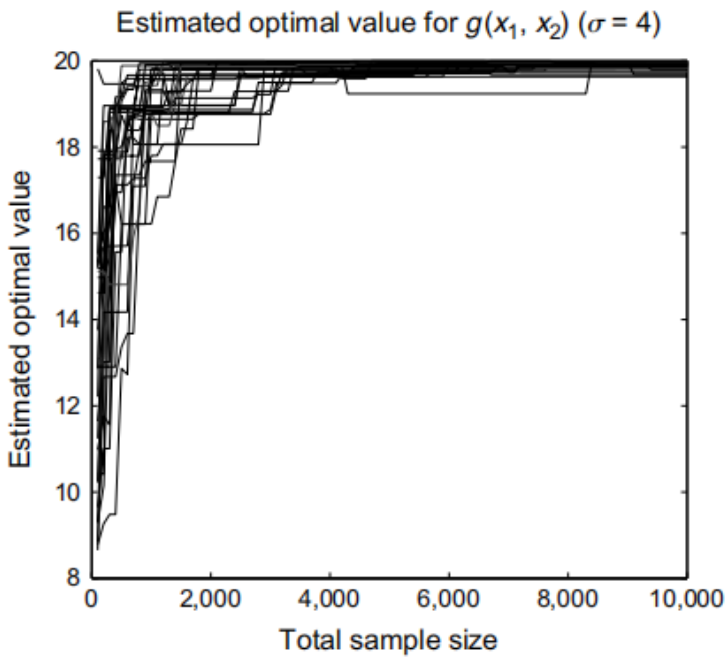
A Problem with Multiple Local Optima

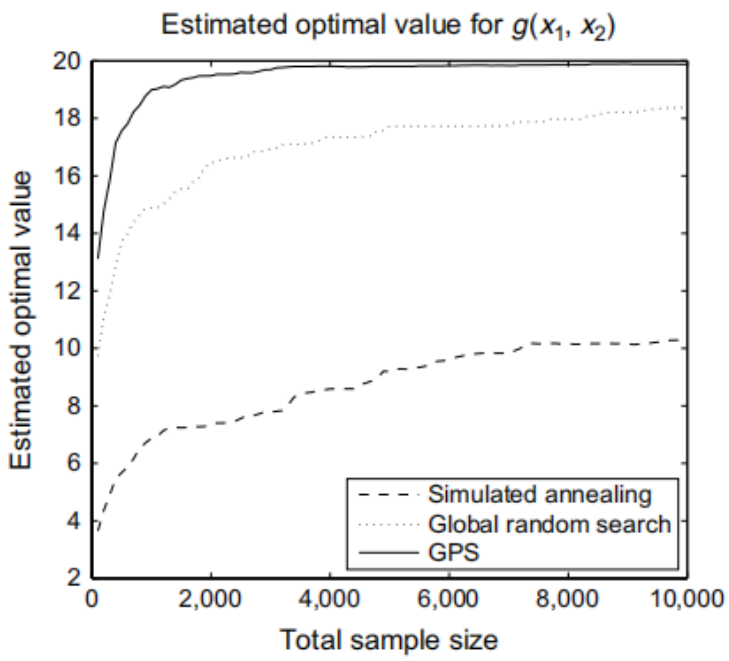


- The function has 10^8 feasible solutions and 25 local optimal solutions.
- The values of the three highest peaks are 20, 19.17 and 19.17, respectively
- The noise term is $N(0, 3)$.

A Problem with Multiple Local Optima

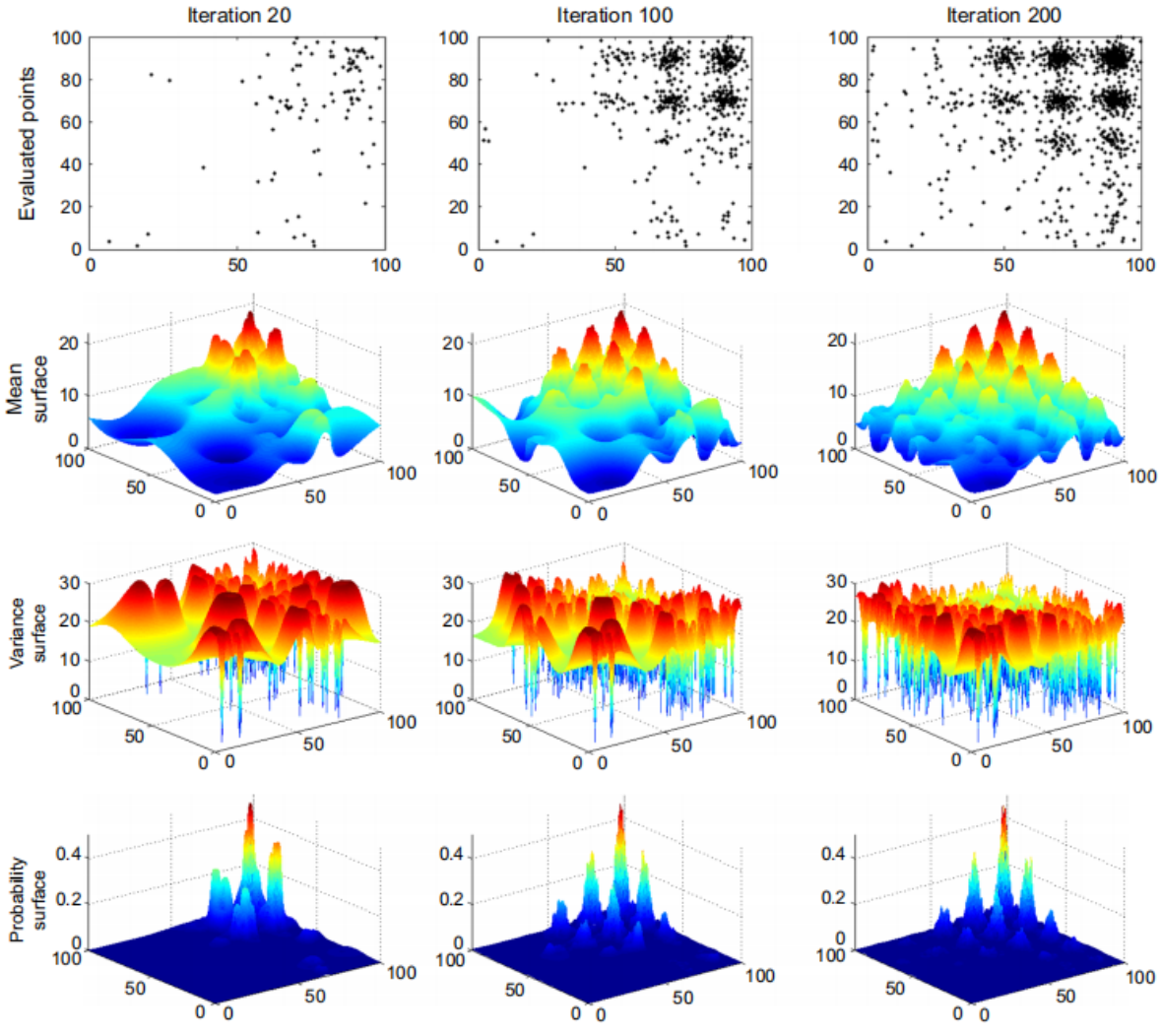
Estimated optimal value for function g.





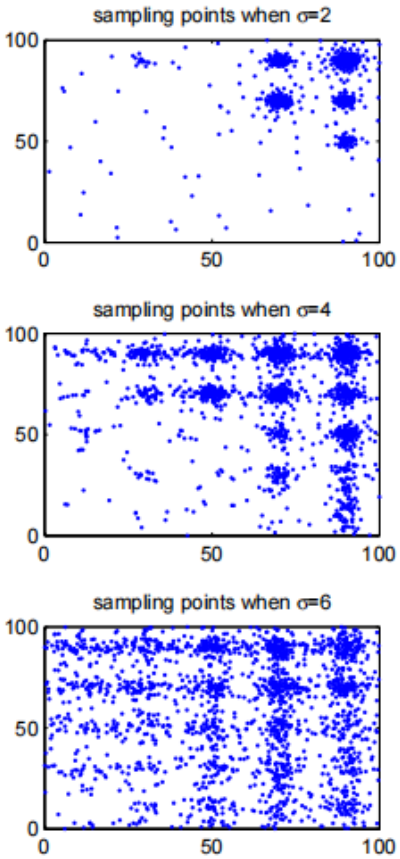
A Problem with Multiple Local Optima

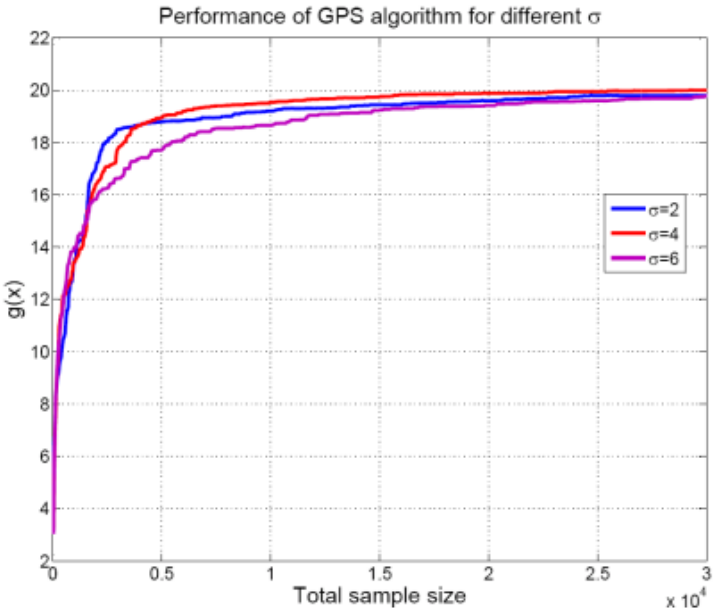
(Color online) The evaluated points, mean surface, variance surface, and probability surface by 20, 100, 200 iterations (approximately 100, 500, and 1,000 evaluated points) when solving Problem (14).



A Problem with Multiple Local Optima

σ is a parameter in the Gaussian process to model the inherent fluctuations of the surface. One may use it to adjust the behavior of the GPS algorithm.





Small σ puts more effort in exploitation, while large σ puts more effort in exploration.

Conclusions

- This paper propose a Gaussian process-based approach to constructing sampling distributions that balance the exploitation and exploration trade-off in a seamless way.
- Develop the GPS algorithm that implements the sampling distributions, analyze its global convergence, and study its practical performances on the numerical examples
- Propose several stopping criteria that may be used in the GPS algorithm and study their numerical performances.

Extension

- First, the sampling distribution can be extended easily to continuous OvS problems. However, the global convergence of the resulted algorithm may be difficult to prove.
- Second,it may be beneficial to adopt a dynamic estimation scheme in the algorithm to improve the finite time performance.
- Third, the issue of designing stopping criteria for globally convergent random search algorithms is both interesting and important and deserves more study.

Refercence

Ankenman B, Nelson BL, Staum J
Stochastic kriging for simulation metamodeling.
Oper. Res. 58(2):371–382,2010

Baumert S, Ghate A, Kiatsupaibul S, Shen Y, Smith RL, Zabinsky ZB J
Discrete hit-and-run for sampling points from arbitrary distributions over subsets of integer hyperrectangles..
Oper. Res. 57(3):727–739.,2009

Thanks for listening !