The Exploration and Exploitation Tradeoff in Discrete Optimization via Simulation

L. Jeff Hong

Department of Industrial Engineering and Logistics Management
The Hong Kong University of Science and Technology

May 2010

Problem Statement

The discrete optimization via simulation (DOvS) problem:

$$\max \; g(x) \quad \text{s.t. } x \in \mathbb{X}$$

- $\mathbb{X}$ is often a finite subset of $\mathbb{Z}^d$.
- There is no closed-form expression of $g(\cdot)$.
- Deterministic simulation: $g(\cdot)$ may be evaluated without noise by running a deterministic simulation experiment, e.g. finite-element analysis.
- Stochastic simulation: $g(x) = \mathrm{E}[G(x)]$, and i.i.d. observations of $G(x)$ may be obtained by running stochastic simulation experiments, e.g. discrete-event simulation.

Random Search Algorithms

- Relaxations of integrality constraints, e.g., branch-and-bound algorithms, cannot be applied, because $g(x)$ cannot be evaluated at non-integer solutions.

- Random search algorithms dominate the literature, e.g.,
  - Stochastic ruler (Yan & Mukai 1992)
  - Stochastic comparison (Gong, Ho and Zhai 1999)
  - Simulated annealing (Alrefaei & Andradóttir 1999)
  - Pure adaptive search (Patel, Smith & Zabinsky 1988)
  - Nested partitions (Shi & Ólafsson 2000, Pichitlamken & Nelson 2003)
  - Random search (Andradóttir 1995 & 1996)
  - COMPASS (Hong & Nelson 2006)
  - Industrial strength COMPASS (Xu, Nelson and Hong 2010)
  - MRAS (Hu, Fu and Marcus 2007)

Random Search Algorithms (cont'd)

- Basic framework:

  At iteration k:

  Step 1 (Sampling): Determine a sample distribution over $\mathbb{X}$, denoted as $f_k(x|\mathcal{F}_{k-1})$. Sample a set of solutions based on $f_k(\cdot)$.

  Step 2 (Evaluation): Evaluate (i.e., simulate) the solutions and determine x_k. Let $k = k + 1$.

  $*$ Some algorithms take several rounds of steps 1 and 2 to determine x_k.

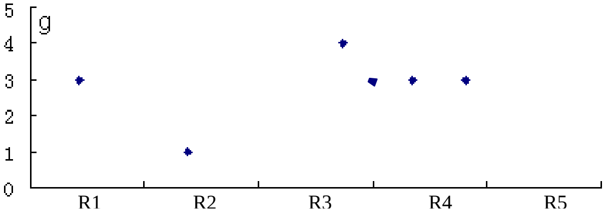
- In this talk, we focus on step 1, which determines a sample distribution $f_k(x|\mathcal{F}_{k-1})$.

Exploration and Exploitation Tradeoff

- Suppose we do not know the convexity of $g(\cdot)$.
- $g(\cdot)$ has some sort of continuity, i.e., solutions that are close to each others tend to have similar objective values.
- To find a better solution, one may search the largely unknown region (exploration, global search) or search around the current solution (exploitation, local search).
- There is a tradeoff between exploration and exploitation in determining the sampling distribution $f_k(x|\mathcal{F}_{k-1})$.

Exploration and Exploitation Tradeoff (cont'd)

Consider a one-dimensional problem where $g(x)$ can be evaluated without noise. Suppose that we are at iteration k and x_{k-1} is the current best solution.

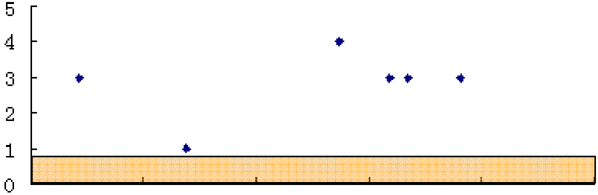


Which region should have more sampling probability?

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

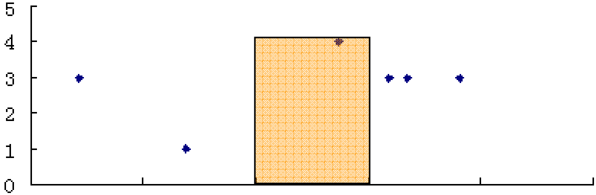
Exploration-based

- Sample all solutions in $\mathbb{X}$ with equal probability.
  - Pure Random Search
  - Global Search Method (Andradóttir 1996).



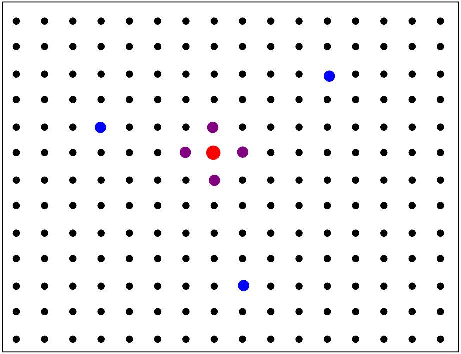
Exploitation-based

- Only sample the solutions in a local neighborhood of current solution.
- Depending if a worse solution can be accepted or not
  - globally-convergent algorithms, e.g. stochastic ruler (Yan and Mukai 1992), stochastic comparison (Gong et al. 1999) and simulated annealing (Alrefaei and Andradóttir 1999) etc.
  - locally-convergent algorithms, e.g. random search (Andradóttir 1995) and COMPASS (Hong and Nelson 2006).



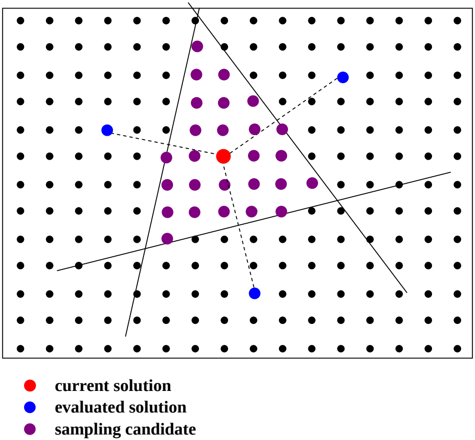
Exploitation-based (cont'd)

Simulated Annealing Algorithm:



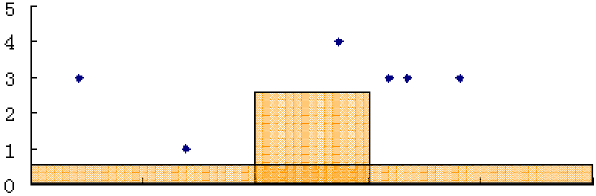
- current solution
- evaluated solution
- sampling candidate

COMPASS Algorithm:



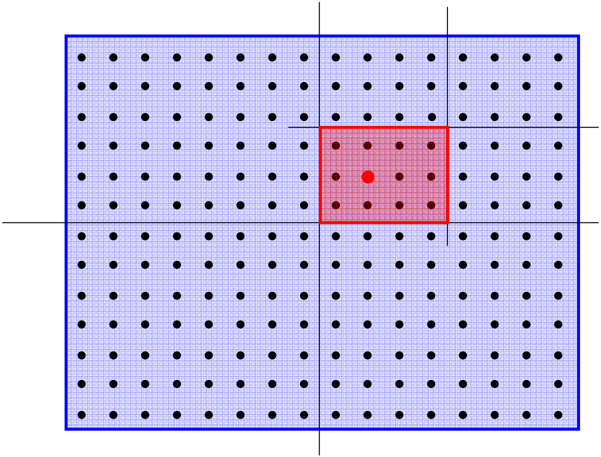
Combined Exploration and Exploitation

- Sampling distribution has two components, one for the local neighborhood, one for the entire region.
  - Nested partitions (Shi and Ólafsson 2000)
  - R-BEES, R-BEESE (Andradóttir and Prudius 2009)
- Some iterations sample from the local neighborhood and others sample from the entire region
  - A-BEES, A-BEESE (Andradóttir and Prudis 2009)



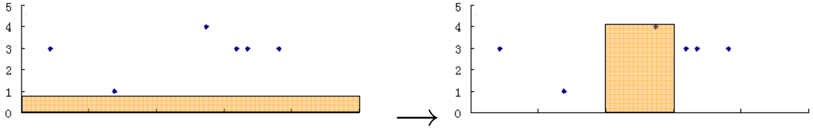
Combined Exploration and Exploitation (cont'd)

Nested Partitions Algorithm:



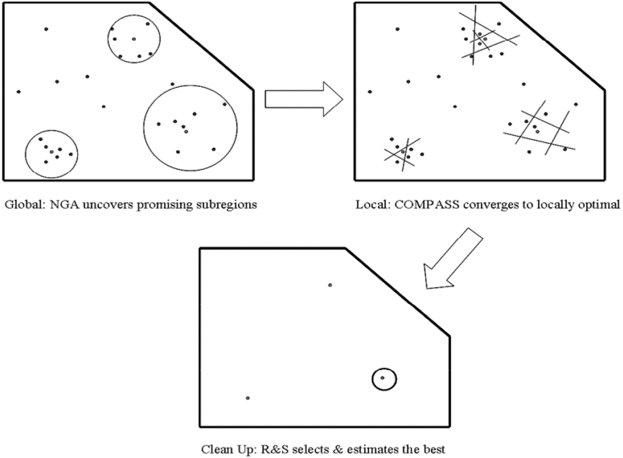
Combined Exploration and Exploitation (cont'd)

- Phases-based, first exploration then exploitation
  - Industrial strength COMPASS (Xu, Nelson and Hong 2010)
- Imbedding a greedy-based exploitation in random search algorithms
  - Pichitlamken and Nelson (2003) added a hill climbing component in each iteration of the Nested Partitions algorithm



Combined Exploration and Exploitation (cont'd)

Industrial Strength COMPASS:



Model-Based Exploration and Exploitation

- These algorithms are typically proposed for continuous simulation optimization problems. However, they are also applicable to discrete problems.
- Directly modeling the sampling distribution
  - MRAS (Hu, Fu and Marcus 2007) has a model of sampling distribution. It updates the sampling distribution based on elite samples in each iteration.
- Response surface methodology
  - Barton and Mechesheimer (2006) provided a nice review on the topic.
  - Kleijnen et al. (many) proposed using kriging to give a fit the function and predict the location of the optimal point from the fitted surface.
  - Powell (2002), Deng and Ferris (2009) and Chang, Hong and Wan (2010) proposed to use an iterative quadratic surface fitting to find a (local) optimal solution.

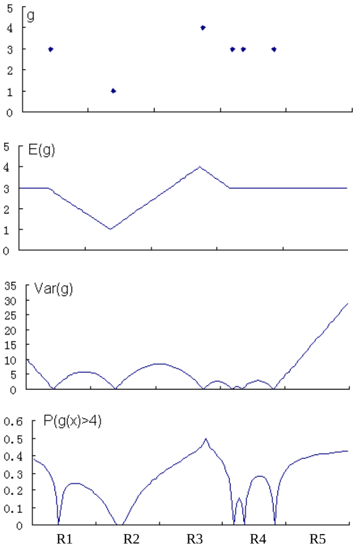
Model-Based Exploration and Exploitation (cont'd)

- Kriging (Gaussian process)-based convergent algorithms:
  - Once a Gaussian process is fitted for the response surface, the distributions of the values of all solutions can be derived under a Bayesian framework;
  - This information may be used to determine the next solution to evaluate.
    - The P-algorithm (Kushner 1964, Torn and Zilinskas 1989) finds the solution that has a highest probability being better than the current best solution by a threshold.
    - the Expected improvement algorithm (Jones, Schoulau and Welch 1998) finds the solution that has a highest expected improvement.
    - Recently, Scott, Frazier and Powell (2010) proposed to use knowledge gradient, which measures the marginal information gain of evaluating a new solution, to determine what solution to evaluate.
    - These algorithms are typically not random search algorithms.

Kriging-based Iterative Random Search Algorithm

Lihua (Lily) Sun & L. Jeff Hong

A Brownian Motion Based Approach for One-dim Problem



R2 < R3, R2 < R5, R1> R4

Kriging Metamodeling

Assuming that $g(x)$ is a sample path of the following Gaussian process

$$Y(x) = u + M(x)$$

where u is a constant, $M(x)$ is a Gaussian process with mean 0 and stationary covariance function $\sigma^2\gamma(\cdot)$, where

$$\gamma(x_1, x_2) = \mathrm{Corr}(M(x_1), M(x_2))$$

and $\gamma(x_1, x_2)$ is typically defined using $\|x_1 - x_2\|$, e.g.,

$$\gamma(x_1, x_2) = \exp(-\rho\|x_1 - x_2\|^2).$$

Once we have observed $g(x_1), \ldots, g(x_n)$, we know that $Y(x)$ goes through $(x_1, g_1), \ldots, (x_n, g_n)$. Then, we can use the Gaussian process to predict the distribution of $g(x)$ at any unknown x.

Kriging Metamodeling (cont'd)

Let $\mathbf{g} = (g_1, \ldots, g_n)'$, $\Gamma = [\gamma(x_i, x_j)]$ which is an $n \times n$ matrix, and $\gamma(x) = (\gamma(x, x_1), \ldots, \gamma(x, x_n))'$ for any $x \in \mathbb{X}$.

Then, the kriging model (typically) predicts

$$\mathrm{E}[g(x)] = \lambda(x)'\mathbf{g}, \quad \lambda(x)' = \left[\gamma(x) + \mathbf{1}\frac{1 - \mathbf{1}'\Gamma^{-1}\gamma(x)}{\mathbf{1}'\Gamma^{-1}\mathbf{1}}\right]' \Gamma^{-1},$$

$$\mathrm{Var}(g(x)) = \sigma^2 \left[\gamma(x)'\Gamma^{-1}\gamma(x) - \frac{(\mathbf{1}'\Gamma^{-1}\gamma(x) - 1)^2}{\mathbf{1}'\Gamma^{-1}\mathbf{1}}\right].$$

and $g(x)$ follows a normal distribution.

Given the distribution $g(x)$, we can calculate $\Pr\{g(x) > g(x_{k-1})\}$ for any $x \in \mathbb{X}$. We can then normalize the probabilities to determine sampling distribution at iteration k.

Kriging Metamodeling (cont'd)

When number of points becomes large, e.g., $n \geq 500$,

- inverting Γ is computationally slow,
- Γ is often ill-conditioned.

Re-examining the choice of $\lambda(x)$:

- $\lambda(x)$ minimizes the mean squared error of estimating $g(x)$,
- $\lambda(x)$ satisfies the following properites:
  - $\mathrm{E}[g(x)]$ is a linear combination of g_i, i.e., $\sum_{i=1}^{n} \lambda_i(x) = 1$,
  - $\lim_{x \to x_i} \mathrm{E}[g(x)] = \mathrm{E}[g(x_i)]$, i.e., $\lim_{x \to x_i} \lambda_j(x) = \delta_{ij}$ where $\delta_{ij} = 1_{\{i=j\}}$,
  - $\lim_{x \to x_i} \mathrm{Var}(g(x)) = 0$.

In random search algorithms, fitting is not so important. The important is to efficiently generate a sampling distribution that balances exploration and exploitation.

Our Kriging-based Framework

We assume $g(x)$ is a sample path of the following process

$$Y(x) = Z(x) + \lambda(x)'(\mathbf{g} - \mathbf{Z}),$$

where $Z(x)$ is an (unconditioned) stationary Gaussian process and $\mathbf{Z} = (Z(x_1), \ldots, Z(x_n))'$.

Condition 1

- $\lambda_i(x) \geq 0$ and $\sum_{i=1}^{n} \lambda_i(x) = 1$;
- $\lambda_i(x_j) = \delta_{ij}$ and $\lim_{x \to x_i} \lambda_j(x) \to \delta_{ij}$.

Under **Condition 1**,

- $Y(x_i) = g_i$;
- $\mathrm{E}[Y(x)] = \lambda(x)'\mathbf{g}$ and $\mathrm{Var}[Y(x)] = \sigma^2 \left(1 - 2\lambda(x)'\gamma(x) + \lambda(x)'\Gamma\lambda(x)\right)$;
- $\lim_{x \to x_i} \mathrm{E}[Y(x)] = g_i$ and $\lim_{x \to x_i} \mathrm{Var}[Y(x_i)] \to 0$.

Our Kriging-based Framework (cont'd)

There are many $\lambda_i(x)$ satisfy **Condition 1**. For instance, we may let

$$\lambda_i(x) = \frac{[1-\gamma(x,x_i)]^{-1}}{\sum_{j=1}^{n}[1-\gamma(x,x_j)]^{-1}}$$

when $\gamma(x_1,x_2) = \exp(-\rho\|x_1 - x_2\|^2)$.

Then, we set the sampling distribution as

$$f_k(x) = \frac{\Pr\left\{Y(x) > g_{k-1}^*\right\}}{\sum_{z\in\mathbb{X}}\Pr\left\{Y(z) > g_{k-1}^*\right\}} \quad \forall x \in \mathbb{X}.$$

Currently, we use an acceptance-rejection algorithm to sample from this distribution. It becomes slow when $n \geq 1000$. We are working on improving this now.

Convergence Property

Condition 2: For any $x_1, , x_2 \in \mathbb{X}$, $\gamma(x_1, x_2) = h(\|x_1 - x_2\|) \geq 0$, where $h(\cdot)$ is a decreasing function, and for any x_0, x_1, x_2, $h(\|x_1 - x_2\|) \geq h(\|x_0 - x_1\|) \cdot h(\|x_0 - x_2\|)$.

Theorem

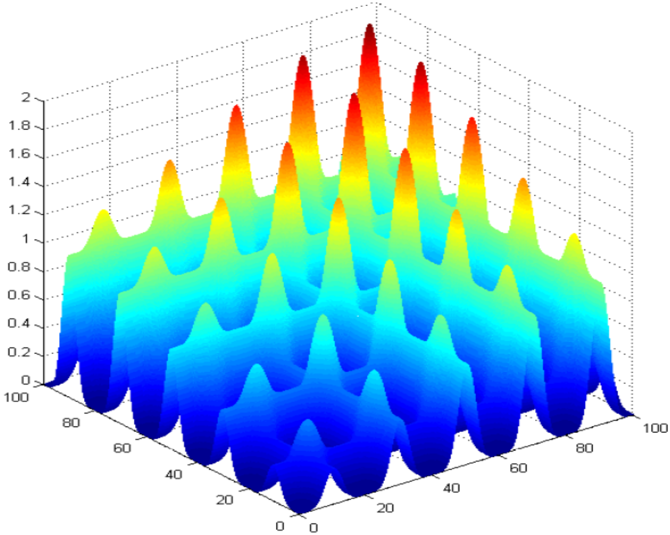
Suppose Conditions 1 and 2 are satisfied. Then,

$$\lim_{k \to \infty} g_k^* = g^*$$

in probability.

*The convergence result holds for continuous problems as well, where $\mathbb{X} \subset \Re^d$.

Numerical Experiments

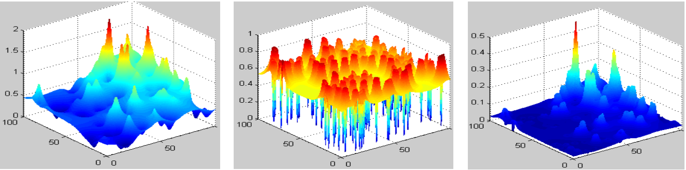


For numerical example, the function value is xi=0.01k, for k=1 to 10000

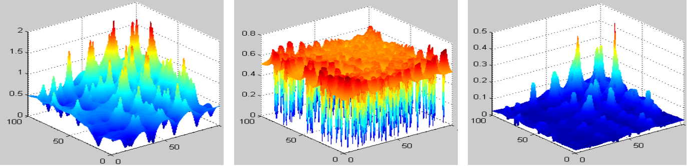
Numerical Experiments (cont'd)

Mean (the fitted surface) | **Variance** | **P(g(x)>g_k^*)**

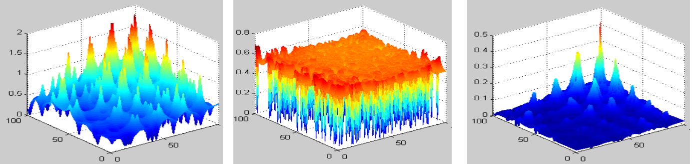
100 iterations



300 iterations

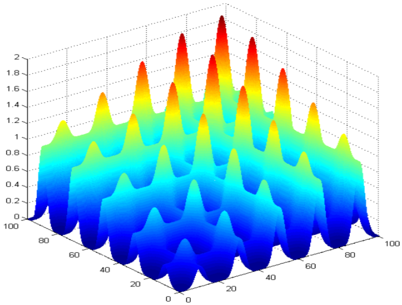


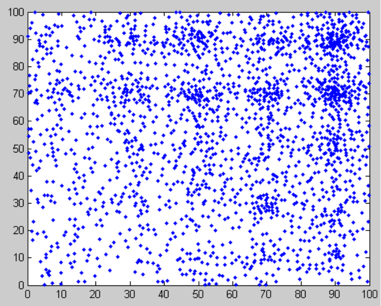
500 iterations



Numerical Experiments (cont'd)

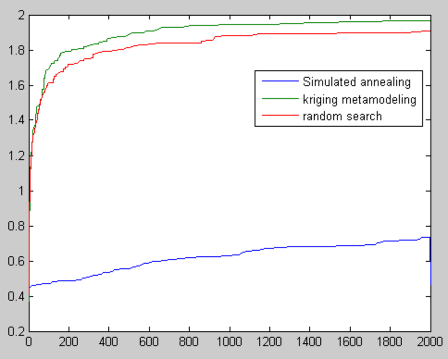
Points sampled by the algorithm:





Numerical Experiments (cont'd)

Comparing to to pure random search and simulated annealing (average of 30 replications)



Extensions

- For stochastic simulation optimization problems, estimation errors need to be considered;
- Some commonly used approach to handling estimation errors:
  - using an increasing number of observations (Hong and Nelson 2006),
  - using an increasing number of comparison (Gong, Ho and Zhai 1999),
  - re-simulating some old solutions (Andradóttir and Prudius 2009)
- In our kriging-based framework, the sampling distribution becomes

$$Y(x) = Z(x) + \lambda(x)'(\bar{\mathbf{g}} - \mathbf{Z}) + \lambda(x)'\boldsymbol{\epsilon},$$

where $\bar{\mathbf{g}} = (\bar{g}(x_1), \ldots, \bar{g}(x_n))'$, and $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)$ where $\epsilon_1, \ldots, \epsilon_n$ are n independent normal random variables with mean 0 and variance $\hat{\sigma}^2(x_i)$. We re-simulate some old elite solutions to remove estimation errors.

Research Questions

- Exploration and exploitation tradeoff exists in many other related areas, e.g., machine learning, approximate dynamic programming;
- Gaussian process is very attractive in fitting global surface and should be studied more for simulation optimization;
- When a Gaussian process is available, shall we do random search (determining a sampling distribution and sampling points randomly) or shall we do deterministic search (determining the point that maximizes probability of better than current point, or expected improvement, or knowledge gradient)?
- Shall we distinguish expensive simulation and not-so-expensive simulation?