Adaptation of the UOBYQA Algorithm for Noisy Functions

Geng Deng   Michael C. Ferris

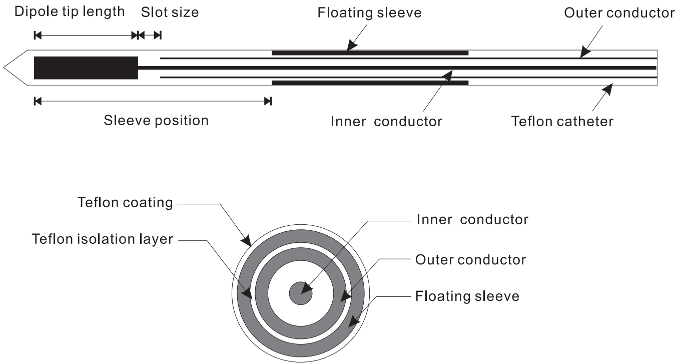
University of Wisconsin-Madison

Winter Simulation Conference '06

Simulation-based optimization problem

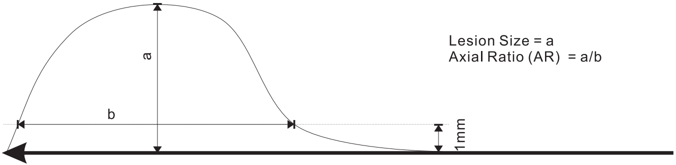
- Computer simulations are used as substitute to evaluate complex real systems.
- Simulations are widely applied in engineering design, manufacturing, supply chain management, medical treatment and many other fields.
- The goal: Optimization finds the best values of the decision variables (design parameters or controls) that minimize some performance measure of the simulation.

Design a coaxial antenna for hepatic tumor ablation



Simulation of the electromagnetic radiation profile

Finite element models (MultiPhysics v3.2) are used to generate the electromagnetic (EM) radiation fields in liver given a particular design



Metric	Measure of	Goal
Lesion radius	Size of lesion in radial direction	Maximize
Axial ratio	Proximity of lesion shape to a sphere	Fit to 0.5
S_{11}	Tail reflection of antenna	Minimize

A general problem formulation

- We formulate the simulation-based optimization problem as

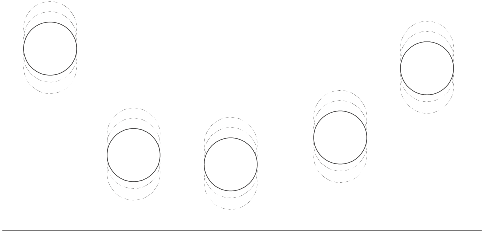
$$\min_{x \in \mathcal{S}} F(x) = \mathbb{E}_\omega[f(x, \omega(x))], \tag{1}$$

where $\omega(x)$ is a random factor arising in the simulation process.
The sample response function $f(x, \omega)$
  - typically does not have a closed form, thus cannot provide gradient or Hessian information
  - is normally computationally expensive
  - is affected by uncertain factors in simulation

The underlying objective function $F(x)$ has to be estimated; for example, by averaging Monte Carlo samples.

The discrete optimization case

- A fundamental step for continuous optimization algorithm design.
- For example, test elasticity of a set of balls. Here $\mathcal{S} = \{1, 2, 3, 4, 5\}$ represents a set of 5 balls.



- Objective: Choose the ball with the largest expected bounce height $F(x_i)$. $f(x_i, \omega_j)$ corresponds to a single measurement in an experiment.

How to select the best system

- First choose the maximum sample mean

$$\arg\max_{i \in \mathcal{S}} \bar{\mu}_i := \frac{1}{N_i} \sum_{j=1}^{N_i} f(x_i, \omega_j), \tag{2}$$

  where N_i is the number of experiments.
- Select the best system with high accuracy, while controlling the total amount of simulation runs.
- Two approaches
  - Indifference zone ranking and selection
    S.-H. Kim and B. L. Nelson, "Selecting the Best System: Theory and Methods."
  - Bayesian approach
    S. E. Chick, and K. Inoue, "New Two-stage and Sequential Procedures for Selecting the Best Simulated System."
    H.-C. Chen, C.-H. Chen, and E. Yucesan, "An Asymptotic Allocation for Simultaneous Simulation Experiments."

Bayesian approach

- Denote the mean of the simulation output for each system as

$$\mu_i = F(x_i) = \mathbb{E}_\omega[f(x_i, \omega)].$$

- In Bayesian perspective, the means are considered as Gaussian random variables whose posterior distributions can be estimated as

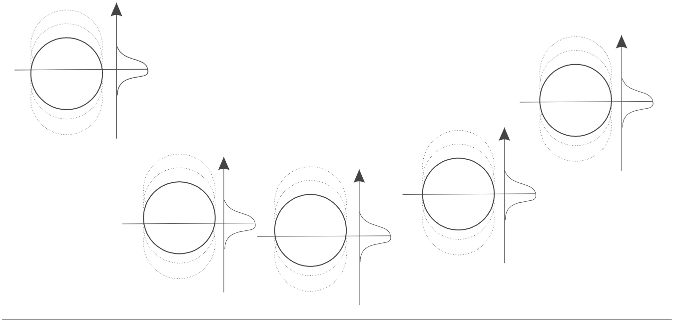
$$\mu_i | X \sim N(\bar{\mu}_i, \hat{\sigma}_i^2 / N_i), \tag{3}$$

where $\bar{\mu}_i$ is sample mean and $\hat{\sigma}_i^2$ is sample variance.

- We can derive other types of posterior distributions. The above Gaussian formulation is easy to manipulate, and is guaranteed by Central Limit Theorem.

Posterior distributions facilitate comparison

Select the first ball



Now it is easy to compute the probability of correct selection (PCS).

Compute the PCS

- Pairwise comparison

$$PCS = Pr(\mu_1 \geq \mu_2) \sim Pr(\mu_1 \geq \mu_2 | X) = Pr(\mu_1|X - \mu_2|X \geq 0). \tag{4}$$

- Multiple comparisons (Bonferroni inequality):

$$\begin{aligned} PCS &= Pr(\mu_b - \mu_i \geq 0, i = \{1, 2, \cdots, K\} \setminus \{b\}) \\ &\sim 1 - \textstyle\sum_{i=1, i \neq b}^{K} Pr(\mu_b - \mu_i < 0). \end{aligned} \tag{5}$$

Summary of the Bayesian approach

- Once the PCS is determined, future work is to choose the suitable sample number of each system N_i such that the best system is selected with desired accuracy

$$PCS \geq 1 - \alpha.$$

- Issues concerning how to optimally allocate computational resources.
- Bayesian approach
  - utilizes both mean and variance information
  - simple and direct to implement
  - without using indifference-zone parameter δ

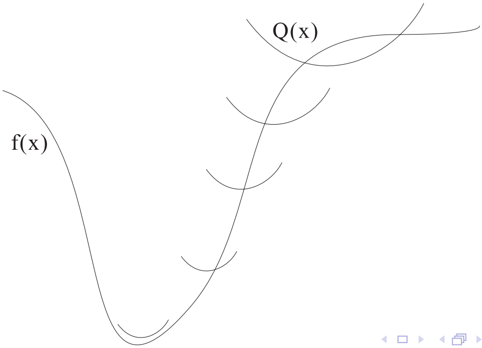
Unconstrained continuous optimization case

$\mathcal{S} = \mathbb{R}^n$

- Basic approach: reduce function uncertainty by averaging multiple samples per point, which is similar to the discrete case.
- Potential difficulty:
  efficiency of algorithm VS number of simulation runs
- We apply Bayesian approach to determine appropriate number of samples per point, while simultaneously enhancing the algorithm efficiency
- Guarantee the global convergence of the algorithm

Noisy UOBYQA: a noisy extension of the UOBYQA algorithm

The base derivative free optimization algorithm: The UOBYQA algorithm (Unconstrained Optimization BY Quadratic Approximation) is based on a trust region method. It constructs a series of local quadratic approximation models of the underlying function.



Quadratic model construction and solve trust region subproblem

(a) construct a quadratic model via interpolation

$$Q(x,\omega) = f(x_k,\omega) + g_Q^T(\omega)(x-x_k) + \frac{1}{2}(x-x_k)^T G_Q(\omega)(x-x_k) \quad (6)$$

The model is unstable since interpolating noisy data

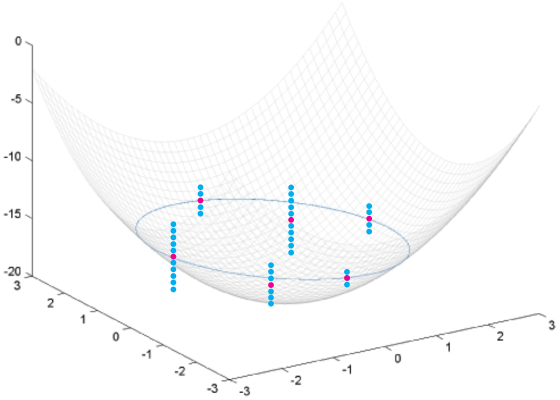
(b) Solve trust region subproblem

$$\begin{aligned} s_k(\omega) = \arg\min_s \quad & Q(x_k + s, \omega) \\ s.t. \quad & \|s\|_2 \leq \Delta_k \end{aligned} \quad (7)$$

The solution is thus unstable

(e) Update a new iterate x_{k+1} by comparing function values $f(x_k)$ and $f(x_k + s_k^*)$. Use pairwise comparison

Why is the quadratic model unstable?



How to stabilize the quadratic model?

Let $\mathcal{I} = \{y^1, y^2, \ldots, y^L\}$ be the interpolation set.

- Quadratic interpolation model is a linear combination of Lagrange functions:

$$Q(x,\omega) = \sum_{j=1}^{L} f(y^j,\omega) l_j(x). \tag{8}$$

- Each piece $l_j(x)$ is a quadratic polynomial, satisfying

$$l_j(y^i) = \delta_{ij}, i = 1, 2, \cdots, L.$$

- The coefficients of l_j are uniquely determined, regardless of the random objective function.

Bayesian estimation of coefficients c_Q, g_Q, G_Q

In Bayesian approach, the mean of function output $\mu(y^j) := \mathbb{E}_\omega f(y^j, \omega)$ is considered as a random variable: Normal posterior distributions:

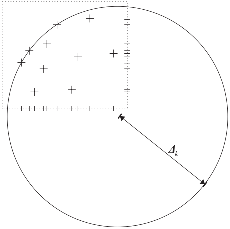
$$\mu(y^j)|X \sim N(\bar{\mu}(y^j), \hat{\sigma}^2(y^j)/N_j). \tag{9}$$

Thus the coefficients of the quadratic model are estimated as:

$$\begin{aligned} g_Q|X &= \textstyle\sum_{j=1}^{L}(\mu(y^j)|X)g_j, \\ G_Q|X &= \textstyle\sum_{j=1}^{L}(\mu(y^j)|X)G_j. \end{aligned} \tag{10}$$

- g_j, G_j are coefficients of Lagrange functions l_j.
- g_j, G_j are deterministic and determined by points y^j.

Constraining the variance of coefficients

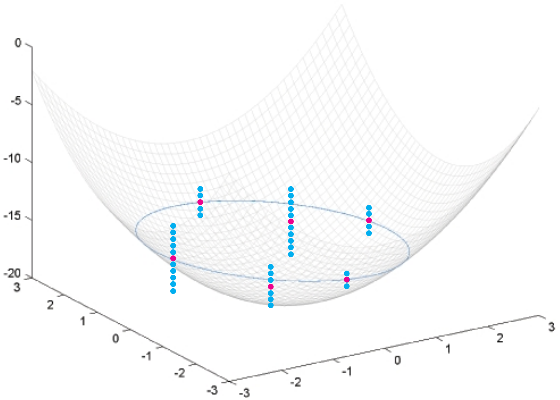


- Generate samples of function values from these (estimated) distributions.
- Trial solutions are generated within a trust region. The standard deviation of the solutions are constrained.

$$\max_{i=1}^{n} std([s^{*(1)}(i), s^{*(2)}(i), \cdots, s^{*(M)}(i)]) \leq \beta\Delta_k. \qquad (11)$$

Optimally allocating computing resources

Select appropriate N_j for the point y^j in the interpolation set



Computational issues

- Allocation of computational resources is determined by:

$$\frac{std(g_Q(i'))}{\mathbb{E}[g_Q(i')]} \leq \beta, i' = 1, \cdots, n \quad (12)$$

$$\frac{std(G_Q(i',j'))}{\mathbb{E}[G_Q(i',j')]} \leq \beta, i', j' = 1, \cdots, n \quad (13)$$

- Compare two points x_k and $x_k + s_k^*$ using pairwise comparison. The new iterate is set as the better point. (refer to previous slides)
- New termination criterion to stop the algorithm appropriately.

A numerical test

Table: Noisy UOBYQA for the Rosenbrock function, $n = 2$ and $\sigma^2 = 0.01$.

Iteration (k)	FN	$F(x_k)$	Δ_k
1	1	404	2
20	78	3.56	9.8×10^{-1}
40	140	0.75	1.2×10^{-1}
60	580	0.10	4.5×10^{-2}
80	786	0.0017	5.2×10^{-3}
100	1254	0.0019	2.8×10^{-4}
120	2003	0.0016	1.1×10^{-4}
✓ Stops here with the termination criterion $\Delta_k \leq 10^{-4}$			

Conclusions

- An efficient, derivative free method for optimizing noisy functions.
- Bayesian techniques applied to balance **efficiency of algorithm VS number of simulation runs**
- The underlying ideas are applicable to many other algorithms.