Some Theory Behind Algorithms for Stochastic Optimization

Zelda Zabinsky

University of Washington
Industrial and Systems Engineering

May 24, 2010

NSF Workshop on Simulation Optimization
Overview

- Problem formulation
- Theoretical performance of stochastic adaptive search methods
- Algorithms based on Hit-and-Run to approximate theoretical performance
- Incorporate random sampling and noisy objective functions
What is Stochastic Optimization?

- Randomness in algorithm AND/OR in function evaluation

- Related terms:
  - Simulation optimization
  - Optimization via simulation
  - Random search methods
  - Stochastic approximation
  - Stochastic programming
  - Design of experiments
  - Response surface optimization
Problem Formulation

- Minimize $f(x)$ subject to $x$ in $S$
- $x$: $n$ variables, continuous and/or discrete
- $f(x)$: objective function, could be black-box, ill-structured, noisy
- $S$: feasible set, nonlinear constraints, or membership oracle
- Assume an optimum $x^*$ exists, with $y^* = f(x^*)$
Example Problem Formulations

- Maximize expected value subject to standard deviation < b
- Minimize standard deviation subject to expected value > t
- Minimize CVaR (conditional value at risk)
- Minimize sum of least squares from data
- Maximize probability of satisfying noisy constraints
Approximate or Estimate $f(x)$?

- Approximate a complicated function:
  - Taylor series expansion
  - Finite element analysis
  - Computational fluid dynamics

- Estimate a noisy function with:
  - Replications
  - Length of discrete-event simulation run
Noisy Objective Function

\[ f(x) = E_\Omega [g(x, \omega_x)] \]
Scenario-based Recourse Function

\[ c^T y + E_\Omega \left[ \min g_y(x, \omega) \right] \]
Local versus Global Optima

- “Local” optima are relative to the neighborhood and algorithm

\[ f(x) \]

\[ x \]

\[ S \]
Local versus Global Optima

- “Local” optima are relative to the neighborhood and algorithm.
Research Question:
What Do We Really Want?

- Do we really just want the optimum?
- What about sensitivity?
- Do we want to approximate the entire surface?
- Multi-criteria?
- Role of objective function and constraints?
- Where does randomness appear?
How can we solve…?

IDEAL Algorithm:
- Optimizes any function quickly and accurately
- Provides information on how “good” the solution is
- Handles black-box and/or noisy functions, with continuous and/or discrete variables
- Is easy to implement and use
Theoretical Performance of Stochastic Adaptive Search

- What kind of performance can we hope for?
- Global optimization problems are NP-hard
- Tradeoff between accuracy and computation
- Sacrifice guarantee of optimality for speed in finding a “good” solution
- Three theoretical constructs:
  - Pure adaptive search (PAS)
  - Hesitant adaptive search (HAS)
  - Annealing adaptive search (AAS)
Performance of Two Simple Methods

- **Grid Search**: Number of grid points is $O((L/\varepsilon)^n)$, where $L$ is the Lipschitz constant, $n$ is the dimension, and $\varepsilon$ is distance to the optimum.

- **Pure Random Search**: Expected number of points is $O(1/p(y^*+\varepsilon))$, where $p(y^*+\varepsilon)$ is the probability of sampling within $\varepsilon$ of the optimum $y^*$.

- Complexity of both is exponential in dimension.
Pure Adaptive Search (PAS)

- PAS: chooses points uniformly distributed in improving level sets
Bounds on Expected Number of Iterations

- PAS (continuous):
  \[ E[N(y^* + \varepsilon)] \leq 1 + \ln \left( \frac{1}{p(y^* + \varepsilon)} \right) \]
  where \( p(y^* + \varepsilon) \) is the probability of PRS sampling within \( \varepsilon \) of the global optimum \( y^* \)

- PAS (finite):
  \[ E[N(y^*)] \leq 1 + \ln \left( \frac{1}{p_1} \right) \]
  where \( p_1 \) is the probability of PRS sampling the global optimum

[Zabinsky and Smith, 1992]
[Zabinsky, Wood, Steel and Baritompa, 1995]
Pure Adaptive Search

- Theoretically, PAS is LINEAR in dimension

**Theorem:**

For any global optimization problem in $n$ dimensions, with Lipschitz constant at most $L$, and convex feasible region with diameter at most $D$, the expected number of PAS points to get within $\varepsilon$ of the global optimum is:

$$E[N(y^*+\varepsilon)] \leq 1 + n \ln(LD / \varepsilon)$$

[Zabinsky and Smith, 1992]
Finite PAS

- Analogous LINEARITY result
- Theorem:
  For an $n$ dimensional lattice $\{1,\ldots,k\}^n$, with distinct objective function values, the expected number of points for PAS, sampling uniformly, to first reach the global optimum is:

$$E[N(y^*)] < 2 + n \ln(k)$$

[Zabinsky, Wood, Steel and Baritompa, 1995]
Hesitant Adaptive Search (HAS)

What if we sample improving level sets with “bettering” probability $b(y)$ and “hesitate” with probability $1 - b(y)$?

$$E[N(y^* + \epsilon)] = \int_{y^* + \epsilon}^{\infty} \frac{d\rho(t)}{b(t)p(t)}$$

where $\rho(t)$ is the underlying sampling distribution and $p(t)$ is the probability of sampling $t$ or better

[Bulger and Wood, 1998]
General HAS

- For a mixed discrete and continuous global optimization problem, the expected value of $N(y^*+\varepsilon)$, the variance, and the complete distribution can be expressed using the sampling distribution $\rho(t)$ and bettering probabilities $b(y)$

[Wood, Zabinsky and Kristinsdottir, 2001]
Annealing Adaptive Search (AAS)

- What if we sample from the original feasible region each iteration, but change distributions?

- Generate points over the whole domain using a Boltzmann distribution parameterized by temperature $T$
  - Boltzmann distribution becomes more concentrated around the global optima as the temperature decreases
  - Temperature is determined by a cooling schedule

- The *record values* of AAS are dominated by PAS and thus LINEAR in dimension

[Romeijn and Smith, 1994]
Performance of Annealing Adaptive Search

- The expected number of sample points of AAS is bounded by HAS with a specific $b(y)$

- Select the next temperature so that the probability of generating an improvement under that Boltzmann distribution is at least $1-\alpha$, i.e.,

$$P\left(Y_{R(k)+1}^{AAS} < y \mid Y_{R(k)}^{AAS} = y\right) \geq 1 - \alpha$$

- Then the expected number of AAS sample points is LINEAR in dimension

  [Shen, Kiatsupaibul, Zabinsky and Smith, 2007]
AAS with Adaptive Cooling Schedule

Y_{record} = f(X_1)
T_1 = \tau(Y_{record}) = 1
T_2 = T_1 = 1

Y_{record} = f(X_3)
T_3 = \tau(Y_{record}) = 0.5

Y_{record} = f(X_0)
T_0 = \tau(Y_{record}) = 10
Research Areas

- Develop theoretical analysis of PAS, HAS, AAS for noisy or approximate functions
  - Model approximation or estimation error
  - Characterize impact of error on performance

- Use theory to develop algorithms
  - Approximate sampling from improving sets (as PAS) or Boltzmann distributions (as AAS)
  - Use HAS, with $\rho(t)$ and $b(y)$, to quantify and balance accuracy and efficiency
Random Search Algorithms

- **Instance-based methods**
  - Sequential random search
  - Multi-start and population-based algorithms

- **Model-based methods**
  - Importance sampling
  - Cross-entropy [Rubinstein and Kroese, 2004]
  - Model reference adaptive search [Hu, Fu and Marcus, 2007]

[Zlochin, Birattari, Meuleau and Dorigo, 2004]
Sequential Random Search

- **Stochastic approximation** [Robbins and Monro, 1951]
- **Step-size algorithms** [Rastrigin, 1960] [Solis and Wets, 1981]
- **Simulated annealing** [Romeijn and Smith, 1994], [Alrafaei and Andradottir, 1999]
- **Tabu search** [Glover and Kochenberger, 2003]
- **Nested partition** [Shi and Olafsson, 2000]
- **COMPASS** [Hong and Nelson, 2006]

- View these algorithms as Markov chains with
  - Candidate point generators
  - Update procedures
Use Hit-and-Run to Approximate AAS

- Hit-and-Run is a Markov chain Monte Carlo (MCMC) sampler
  - converges to a uniform distribution
    [Smith, 1984]
  - in polynomial time $O(n^3)$
    [Lovász, 1999]
  - can approximate any arbitrary distribution by using a filter

- The difficulty of implementing AAS is to generate points directly from a family of Boltzmann distributions
Hit-and-Run

- Hit-and-Run generates a random direction (uniformly distributed on a hypersphere) and a random point (uniformly distributed on the line)
Improving Hit-and-Run

- IHR: choose a random direction and a random point, accept only improving points
Improving Hit-and-Run

- IHR: choose a random direction and a random point, accept only improving points
Improving Hit-and-Run

- IHR: choose a random direction and a random point, accept only improving points
Is IHR Efficient in Dimension?

Theorem:
For any elliptical program in \( n \) dimensions, the expected number of function evaluations for IHR is: \( O(n^{5/2}) \)

[Zabinsky, Smith, McDonald, Romeijn and Kaufman, 1993]
Is IHR Efficient in Dimension?

**Theorem:**
For any elliptical program in \( n \) dimensions, the expected number of function evaluations for IHR is: \( O(n^{5/2}) \)

[Zabinsky, Smith, McDonald, Romeijn and Kaufman, 1993]
Use Hit-and-Run to Approximate Annealing Adaptive Search

- **Hide-and-Seek**: add a probabilistic Metropolis acceptance-rejection criterion to Hit-and-Run to approximate the Boltzmann distribution
  
  [Romeijn and Smith, 1994]

- Converges in probability with almost any cooling schedule driving temperature to zero

- **AAS Adaptive Cooling Schedule**:
  
  - Temperature values according to AAS to maintain $1 - \alpha$ probability of improvement
  
  - Update temperature when record values are obtained

  [Shen, Kiatsupaibul, Zabinsky and Smith, 2007]
Research Possibilities:

- How long should we execute Hit-and-Run at a fixed temperature?
- What is the benefit of sequential temperatures (warm starts) on convergence rate?
- Hit-and-Run has fast convergence on “well-rounded” sets; how can we modify transition kernel in general?
- Incorporate new Hit-and-Run on mixed integer/continuous sets
  - Discrete hit-and-run
    [Baumert, Ghate, Kiatsupaibul, Shen, Smith and Zabinsky, 2009]
  - Pattern hit-and-run
    [Mete, Shen, Zabinsky, Kiatsupaibul and Smith, 2010]
Simulated Annealing with Multi-start: When to Stop or Restart a Run?

- Use HAS to model progress of a heuristic random search algorithm and estimate associated parameters

- **Dynamic Multi-start Sequential Search**
  - If current run appears “stuck” according to HAS analysis, stop and restart
  - Estimate probability of achieving $y^* + \varepsilon$ based on observed values and estimated parameters
  - If probability is high enough, terminate

[Zabinsky, Bulger and Khompatraporn, 2010]
Meta-control of Interacting-Particle Algorithm

- Interacting-Particle Algorithm
  - Combines simulated annealing and population based algorithms
  - Uses statistical physics and Feynman-Kac formulas to develop selection probabilities

[Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, 2004]

- Meta-control approach to dynamically heat and cool temperature

[Kohn, Zabinsky and Brayman, 2006]
[Molvalioglu, Zabinsky and Kohn, 2009]
Meta-control Approach

\[ f(x), S \]

Initial parameters

Interacting-Particle Algorithm

N-Particle Exploration

N-Particle Selection

Control System

State Estimation and System Dynamics

Performance Criterion

Optimal Control Problem

Current Information

Optimal Parameter
Research Possibilities

- Combine theoretical analyses with MCMC and meta-control to:
  - Control the exploration transition probabilities
  - Obtain stopping criterion and quality of solution
  - Relate interacting particles to cloning/splitting

- Combine theoretical analyses and meta-control with model-based approach
Another Research Area: Quantum Global Optimization

- Grover’s Adaptive Search can implement PAS on a quantum computer
  [Baritompa, Bulger and Wood, 2005]

- Apply research on quantum control theory to global optimization
  - [Gardiner, *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, 2004]
  - [Del Moral, *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, 2004]
Optimization of Noisy Functions

- Use random sampling to explore the feasible region and estimate the objective function with replications

- Recognize two sources of noise:
  - Randomness in the sampling distribution
  - Randomness in the objective function

- Adaptively adjust the number of samples and the number of replications
Noisy Objective Function

\[ f(x) \]

[Graph showing a noisy objective function with a blue curve and green error bars, labeled \( f(x) \) on the y-axis and \( x \) on the x-axis.]
Noisy Objective Function

\[ f(x) \]

\[ y(\delta, S) \]

\[ L(\delta, S) \]

\[ \delta \]

\[ x \]
Probabilistic Branch-and-Bound (PBnB)
Sample $N^*$ Uniform Random Points with $R^*$ Replications

$$f(x)$$

$\sigma_1$

$\sigma_2$

$S$

$x$
Use Order Statistics to Assess Range Distribution

\[ f(x) \]

\[ \sigma_1 \]

\[ \sigma_2 \]
Prune, if Statistically Confident

$$f(x)$$

$$\sigma_1$$

$$\sigma_2$$
Subdivide & Sample Additional Points

\[ f(x) \]

\[ \sigma_{11} \quad \sigma_{12} \quad \sigma_2 \]

\[ S \]
Reassess Range Distribution
If No Pruning, Then Continue …
PBnB: Numerical Example
Research Areas

- Develop theory to tradeoff accuracy with computational effort
- Use theory to develop algorithms that give insight into original problem
  - Global optima and sensitivity
  - Shape of the function or range distribution
- Use interdisciplinary approaches to incorporate feedback
Summary

- Theoretical analysis of PAS, HAS, AAS motivates random search algorithms
- Hit-and-Run is a MCMC method to approximate theoretical performance
- Meta-control theory allows adaptation based on observations
- Probabilistic Branch-and-Bound incorporates noisy function evaluations and sampling noise into analysis
Additional Slides for Details on Interacting Particle Algorithm
Interacting-Particle Algorithm

- Simulated Annealing: Markov chain Monte Carlo method for approximating a sequence of Boltzmann distributions

\[ \eta_t(dx) = \frac{e^{-f(x)/T_t}}{\int_S e^{-f(y)/T_t} dy} dx \]

- Population-based Algorithms: simulate a distribution (e.g. Feynman-Kac annealing model) such that

\[ \mathbb{E}_{\eta_t}(f) \to y^* \quad \text{as} \quad t \to \infty \]
Interacting-Particle Algorithm

- **Initialization**: Sample the initial locations $y^i_0$ for particle $i=1,...,N$ from the distribution $\eta_0$

  For $t=0,1,...$,

- **N-particle exploration**: Move particle $i=1,...,N$ to location $\hat{y}^i_t$ with probability distribution $E(y^i_t,d\hat{y}^i_t)$

- **Temperature Parameter Update**:
  
  $T_t = (1 + \varepsilon_t)T_{t-1}, \quad \varepsilon_t \geq -1$

- **N-particle selection**: Set the particles’ locations $y^k_{t+1}, i = 1,...,N$

  $y^k_{t+1} = \hat{y}^i_t$ with probability

  $$\frac{e^{-f(\hat{y}^i_t)/T_t}}{\sum_{j=1}^{N} e^{-f(\hat{y}^j_t)/T_t}}$$
Illustration of Interacting-Particle Algorithm

**Initialization:** Sample $N=10$ points uniformly on $S$

**$N$-particle exploration:**
Move particles from $\bigcirc$ to $\bigstar$ using Markov kernel $E$
Illustration of Interacting-Particle Algorithm

Initialization: Sample N=10 points uniformly on S

N-particle exploration:
Move particles from ◆ to ★ using Markov kernel E
Evaluate $f(.)$

Lower $f(.)$ ★ Higher $f(.)$
Initialization: Sample N=10 points uniformly on $S$

N-particle exploration:
Move particles from $\bigcirc$ to $\star$ using Markov kernel $E$
Evaluate $f(.)$

Lower $f(.)$ $\Leftarrow$ $\Rightarrow$ Higher $f(.)$

N-particle selection:
Move particles from $\star$ to $\bigcirc$ according to their objective function values
Illustration of Interacting-Particle Algorithm

**Initialization:** Sample N=10 points uniformly on S

**N-particle exploration:**
Move particles from \( \bigcirc \) to \( \bigstar \) using Markov kernel \( E \)
Evaluate \( f(.) \)
Lower \( f(.) \) \( \longrightarrow \) Higher \( f(.) \)

**N-particle selection:**
Move particles from \( \bigstar \) to \( \bigcirc \) according to their objective function values
Multi-start or Population-based Algorithms

- Multi-start and clustering algorithms [Rinnooy Kan and Timmer, 1987] [Locatelli and Schoen, 1999]
- Genetic algorithms [Davis, 1991]
- Evolutionary programming [Bäck, Fogel and Michalewicz, 1997]
- Particle swarm optimization [Kennedy, Eberhart and Shi, 2001]
- Interacting particle algorithm [del Moral, 2004]