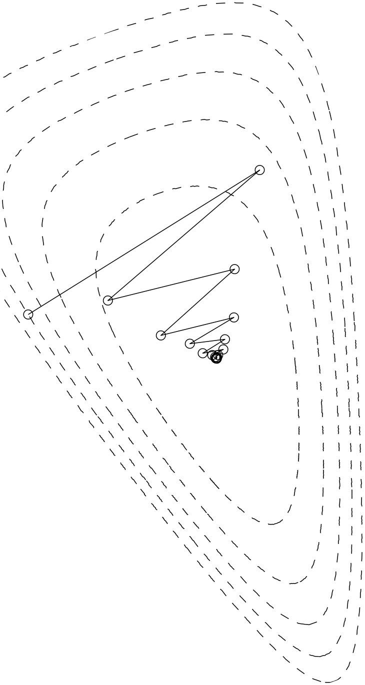


Introduction to Optimization

Blackbox Optimization

Marc Toussaint
U Stuttgart

“Blackbox Optimization”

- The term is not really well defined
  – I use it to express that *only* $f(x)$ can be evaluated
  – $\nabla f(x)$ or $\nabla^2 f(x)$ are not (directly) accessible
  More common terms:

- **Global optimization**
  - This usually emphasizes that methods should not get stuck in local optima
  - Very very interesting domain – close analogies to (active) Machine Learning, bandits, POMDPs, optimal decision making/planning, optimal experimental design
  - Usually mathematically well founded methods

- **Stochastic search** or **Evolutionary Algorithms** or **Local Search**
  - Usually these are local methods (extensions trying to be “more” global)
  - Various interesting heuristics
  - Some of them (implicitly or explicitly) locally approximating gradients or 2nd order models

Blackbox Optimization

- Problem: Let $x \in \mathbb{R}^n$, $f:\ \mathbb{R}^n \to \mathbb{R}$, find

$$\min_x \ f(x)$$

  where we can only evaluate $f(x)$ for any $x \in \mathbb{R}^n$

- A constrained version: Let $x \in \mathbb{R}^n$, $f:\ \mathbb{R}^n \to \mathbb{R}$, $g:\ \mathbb{R}^n \to \{0, 1\}$, find

$$\min_x \ f(x) \quad \text{s.t.} \quad g(x) = 1$$

  where we can only evaluate $f(x)$ and $g(x)$ for any $x \in \mathbb{R}^n$
  I haven't seen much work on this. Would be interesting to consider this more rigorously.

A zoo of approaches

- People with many different backgrounds drawn into this
  Ranging from heuristics and Evolutionary Algorithms to heavy mathematics

  – Evolutionary Algorithms, esp. Evolution Strategies, Covariance Matrix Adaptation, Estimation of Distribution Algorithms
  – Simulated Annealing, Hill Climing, Downhill Simplex
  – local modelling (gradient/Hessian), global modelling

Optimizing and Learning

- Blackbox optimization is often related to learning:
- When we have local a gradient or Hessian, we can take that local information and run – no need to keep track of the history or learn (exception: BFGS)
- In the Blackbox case we have no local information directly accessible $\to$ one needs to account of the history in some way or another to have an idea where to continue search
- "Accounting for the history" very often means learning: Learning a local or global model of f itself, learning which steps have been successful recently (gradient estimation), or which step directions, or other heuristics

Outline

- Stochastic Search
  - A simple framework that many heuristics and local modelling approaches fit in
  - Evolutionary Algorithms, Covariance Matrix Adaptation, EDAs as special case
- Heuristics
  - Simulated Annealing
  - Hill Climing
  - Downhill Simplex
- Global Optimization
  - Framing the big problem: The *optimal solution to optimization*
  - Mentioning very briefly *No Free Lunch* Theorems
  - Greedy approximations, Kriging-type methods

Stochastic Search

Stochastic Search

- The general recipe:
  - The algorithm maintains a probability distribution $p_\theta(x)$
  - In each iteration it takes n samples $\{x_i\}_{i=1}^n \sim p_\theta(x)$
  - Each x_i is evaluated $\rightarrow$ data $\{(x_i, f(x_i))\}_{i=1}^n$
  - That data is used to update θ

- Stochastic Search:

Input: initial parameter θ, function $f(x)$, distribution model $p_\theta(x)$, update heuristic $h(\theta, D)$
Output: final θ and best point x
1: **repeat**
2: Sample $\{x_i\}_{i=1}^n \sim p_\theta(x)$
3: Evaluate samples, $D = \{(x_i, f(x_i))\}_{i=1}^n$
4: Update $\theta \leftarrow h(\theta, D)$
5: **until** θ converges

Stochastic Search

- The parameter θ is the only "knowledge/information" that is being propagated between iterations
  θ encodes what has been learned from the history
  θ defines where to search in the future

- Evolutionary Algorithms: θ is a parent population
  Evolution Strategies: θ defines a Gaussian with mean & variance
  Estimation of Distribution Algorithms: θ are parameters of some distribution model, e.g. Bayesian Network
  Simulated Annealing: θ is the "current point" and a temperature

Example: Gaussian search distribution (μ, λ)-ES

From 1960s/70s. Rechenberg/Schwefel

- Perhaps the simplest type of distribution model

$$\theta = (\hat{x}) , \quad p_t(x) = \mathcal{N}(x|\hat{x}, \sigma^2)$$

a n-dimenstional isotropic Gaussian with fixed deviation σ

- Update heuristic:
  – Given $D = \{(x_i, f(x_i))\}_{i=1}^{\lambda}$, select μ best: $D' = \text{bestOf}_\mu(D)$
  – Compute the new mean $\hat{x}$ from D'

- This algorithm is called "Evolution Strategy (μ, λ)-ES"
  – The Gaussian is meant to represent a "species"
  – λ offspring are generated
  – the best μ selected

Example: "elitarian" selection $(\mu + \lambda)$-ES

- θ also stores the μ best previous points

$$\theta = (\hat{x}, D') \;, \quad p_t(x) = \mathcal{N}(x|\hat{x}, \sigma^2)$$

- The θ update:
  – Select the μ best from $D' \cup D$: $D' = \text{bestOf}_\mu(D' \cup D)$
  – Compute the new mean $\hat{x}$ from D'
- Is called "elitarian" because good parents can survive

- Consider the $(1+1)$-ES: a Hill Climber
- There is considerable theory on convergence of, e.g., $(1+\lambda)$-ES

Evolutionary Algorithms (EAs)

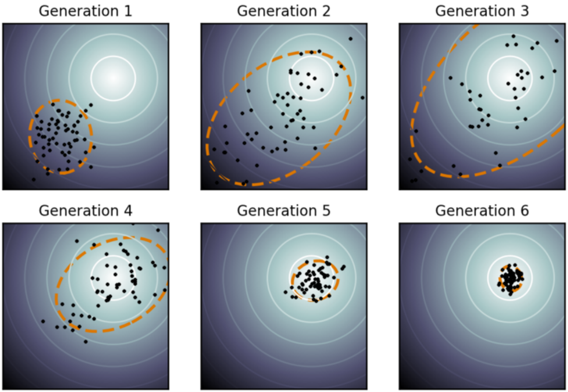
- These were two simple examples of EAs
  Generally, I think EAs can well be described/understood as very special kinds of parameterizing $p_\theta(x)$ and updating θ
  - The θ typically is a set of good points found so far (parents)
  - Mutation & Crossover define $p_\theta(x)$
  - The samples D are called offspring
  - The θ-update is often a selection of the best,
    or "fitness-proportional" or rank-based

- Categories of EAs:
  - **Evolution Strategies**: $x \in \mathbb{R}^n$, often Gaussian $p_\theta(x)$
  - **Genetic Algorithms**: $x \in \{0, 1\}^n$, crossover & mutation define $p_\theta(x)$
  - **Genetic Programming**: x are programs/trees, crossover & mutation
  - **Estimation of Distribution Algorithms**: θ directly defines $p_\theta(x)$

Covariance Matrix Adaptation (CMA-ES)

- An obvious critique of the simple Evolution Strategies:
  - The search distribution $\mathcal{N}(x|\hat{x}, \sigma^2)$ is isotropic (no going *forward*, no preferred direction)
  - The variance σ is fixed!

- Covariance Matrix Adaptation Evolution Strategy (CMA-ES)



Covariance Matrix Adaptation (CMA-ES)

- In Covariance Matrix Adaptation

$$\theta = (\hat{x}, \sigma, C, p_\sigma, p_C) \;, \quad p_\theta(x) = \mathcal{N}(x|\hat{x}, \sigma^2 C)$$

  where C is the covariance matrix of the search distribution
- The θ maintains two more pieces of information: p_σ and p_C capture the "path" (motion) of the mean $\hat{x}$ in recent iterations
- Rough outline of the θ-update:
  – Let $D' = \text{bestOf}_\mu(D)$ be the set of selected points
  – Compute the new mean $\hat{x}$ from D'
  – Update p_σ and p_C proportional to $\hat{x}_{k+1} - \hat{x}_k$
  – Update σ depending on $|p_\sigma|$
  – Update C depending on $p_c p_c^\top$ (rank-1-update) and $\text{Var}(D')$

CMA references

Hansen, N. (2006), "The CMA evolution strategy: a comparing review"
Hansen et al.: Evaluating the CMA Evolution Strategy on Multimodal Test Functions, PPSN 2004.

Function	f_{stop}	init	n	CMA-ES	DE	RES	LOS
$f_{\text{Ackley}}(x)$	1e-3	$[-30, 30]^n$	20	**2667**	.	.	6.0e4
			30	**3701**	12481	1.1e5	9.3e4
			100	**11900**	36801	.	.
$f_{\text{Griewank}}(x)$	1e-3	$[-600, 600]^n$	20	**3111**	8691	.	.
			30	**4455**	11410 *	*8.5e-3/2e5*	.
			100	**12796**	31796	.	.
$f_{\text{Rastrigin}}(x)$	0.9	$[-5.12, 5.12]^n$	20	68586	**12971**	.	9.2e4
		DE: $[-600, 600]^n$	30	147416	**20150** *	1.0e5	2.3e5
			100	1010989	**73620**	.	.
$f_{\text{Rastrigin}}(Ax)$	0.9	$[-5.12, 5.12]^n$	30	**152000**	*171/1.25e6* *	.	.
			100	**1011556**	*944/1.25e6* *	.	.
$f_{\text{Schwefel}}(x)$	1e-3	$[-500, 500]^n$	5	43810	**2567** *	.	7.4e4
			10	240899	**5522** *	.	5.6e5

- For "large enough" populations local minima are avoided

- A variant:
  Igel et al.: A Computational Efficient Covariance Matrix Update and a $(1+1)$-CMA for Evolution Strategies, GECCO 2006.

CMA conclusions

- It is a good starting point for an off-the-shelf blackbox algorithm
- It includes components like estimating the local gradient (p_σ, p_C), the local "Hessian" (Var(D')), smoothing out local minima (large populations)

Estimation of Distribution Algorithms (EDAs)

- Generally, EDAs fit the distribution $p_\theta(x)$ to model the distribution of previously good search points
  For instance, if in all previous distributions, the 3. bit equals the 7. bit, then the search distribution $p_\theta(x)$ should put higher probability on such candidates.
  $p_\theta(x)$ is meant to capture the *structure* in previously good points, i.e. the dependencies/correlation between variables.

- A rather successful class of EDAs on discrete spaces uses graphical models to learn the dependencies between variables, e.g. Bayesian Optimization Algorithm (BOA)

- In continuous domains, CMA is an example for an EDA

Further Ideas

- We could learn a distribution over steps
  – which steps have decreased f recently $\rightarrow$ model
  (Related to “differential evolution”)

- We could learn a distributions over directions only
  $\rightarrow$ sample one $\rightarrow$ line search

Stochastic search conclusions

Input: initial parameter θ, function $f(x)$, distribution model $p_\theta(x)$, update heuristic $h(\theta, D)$

Output: final θ and best point x

```
repeat
    Sample {x_i}_{i=1}^n ~ p_θ(x)
    Evaluate samples, D = {(x_i, f(x_i))}_{i=1}^n
    Update θ ← h(θ, D)
until θ converges
```

- The framework is very general
- The crucial difference between algorithms is their choice of $p_\theta(x)$

Heuristics

- Simulated Annealing
- Hill Climing
- Simplex

Simulated Annealing

- Must read!: An Introduction to MCMC for Machine Learning

Input: initial x, function $f(x)$, proposal distribution $q(x'|x)$
Output: final x

initialilze $T = 1$
repeat
    generate a new sample $x' \sim q(x'|x)$
    acceptance probability $A = \min\left\{1, \frac{e^{-f(x')/T} q(x|x')}{e^{-f(x)/T} q(x'|x)}\right\}$
    With probability A, $x \leftarrow x'$ *// ACCEPT*
    Decrease T
until θ converges

- Typically: $q(x'|x) = \mathcal{N}(x'|x, \sigma^2)$ Gaussian transition probabilities

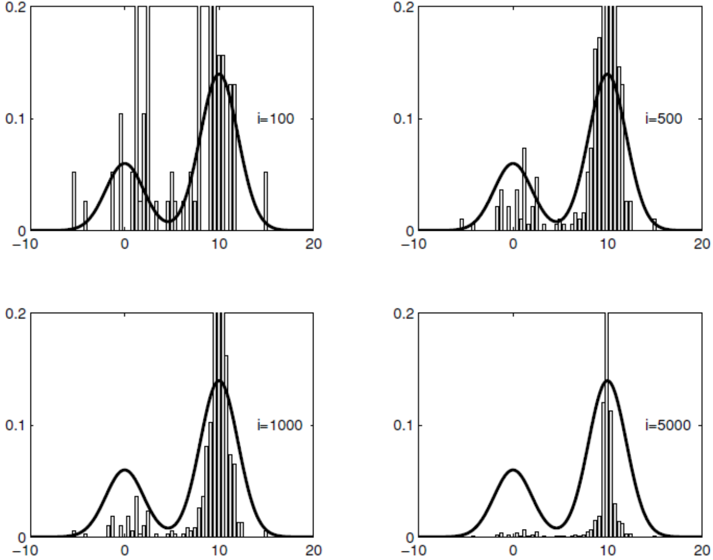
Simulated Annealing

- Simulated Annealing is a Markov chain Monte Carlo (MCMC) method.
- These are iterative methods to sample from a distribution, in our case

$$p(x) \propto e^{-f(x)/T}$$

- For a fixed temperature T, one can show that the set of accepted points is distributed as $p(x)$ (but non-i.i.d.!)
- The acceptance probability compares the $f(x')$ and $f(x)$, but also the reversibility of $q(x'|x)$
- When cooling the temperature, samples focus at the extrema
- Guaranteed to sample all extrema *eventually*

Simulated Annealing



[Wikipedia gif amination]

Hill Climing

- Same as Simulated Annealing with $T = 0$
- Same as $(1+1)$-ES

  There also exists a CMA version of (1+1)-ES, see Igel reference above.

- The role of hill climing should not be underestimated:
  Very often it is efficient to repeat hill climing from many random start points.
- However, no type of learning at all (stepsize, direction)

Nelder-Mead method – Downhill Simplex Method