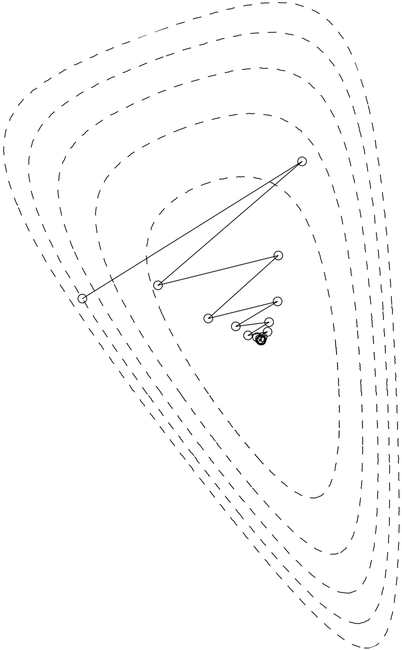


Optimization Algorithms

Stochastic Search & EDAs

Marc Toussaint
Technical University of Berlin
Winter 2024/25

A core aspect in black-box opt is: *What do we estimate from the data?*

- gradient (as in implicit filtering)
- a local model $f_\theta(x)$ (model-based opt.)
- a distribution $p_\theta(x)$ of "good" points (EDAs)

- The θ is what we extract/capture/maintain from the data of previous evaluations

A general stochastic search scheme

- A general stochastic search scheme:
  - The algorithm maintains some information θ
  - This θ defines a *search* distribution $p_\theta(x)$
  - In each iteration it takes λ samples $\{x_i\}_{i=1}^{\lambda} \sim p_\theta(x)$
  - Each x_i is evaluated $\;\rightarrow\;$ new data $D = \{(x_i, f(x_i))\}_{i=1}^{\lambda}$
  - **The new data D is used to update θ**

Input: initial θ, function $f(x)$, distribution model $p_\theta(x)$, update heuristic $h(\theta, D)$
Output: final θ and best point x

repeat
    Sample $\{x_i\}_{i=1}^{\lambda} \sim p_\theta(x)$
    Evaluate samples, $D = \{(x_i, f(x_i))\}_{i=1}^{\lambda}$
    Update $\theta \leftarrow h(\theta, D)$
until θ converges

Evolutionary Algorithms (EAs)

- EAs can well be described as 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 $p_\theta(x)$

Evolution Strategies & EDAs

(as they address continuous optimization in $\mathbb{R}^n$)

Evolution Strategies: Gaussian Search Distribution

[From 1960s/70s. Rechenberg/Schwefel]

- The parameter θ defines a Gaussian search distribution $p_\theta(x)$
- In the simplest case, θ is just the mean $\theta = (\hat{x})$, assuming fixed σ^2:

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

- We sample λ "offspring" $x \sim p_\theta$ to get new data D
- What is a reasonable upate heuristic $\theta \leftarrow h(\theta, D)$?

Evolution Strategies: Gaussian Search Distribution

[From 1960s/70s. Rechenberg/Schwefel]

- The parameter θ defines a Gaussian search distribution $p_\theta(x)$
- In the simplest case, θ is just the mean $\theta = (\hat{x})$, assuming fixed σ^2:

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

- We sample λ "offspring" $x \sim p_\theta$ to get new data D
- What is a reasonable upate heuristic $\theta \leftarrow h(\theta, D)$?
  - **Selection:** Given $D = \{(x_i, f(x_i))\}_{i=1}^{\lambda}$, select the μ best: $D_\mu = \text{bestOf}_\mu(D)$
  - Compute the new mean $\hat{x}$ from D_μ

Evolution Strategies: Gaussian Search Distribution

[From 1960s/70s. Rechenberg/Schwefel]

- The parameter θ defines a Gaussian search distribution $p_\theta(x)$
- In the simplest case, θ is just the mean $\theta = (\hat{x})$, assuming fixed σ^2:

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

- We sample λ “offspring” $x \sim p_\theta$ to get new data D
- What is a reasonable upate heuristic $\theta \leftarrow h(\theta, D)$?
  - **Selection:** Given $D = \{(x_i, f(x_i))\}_{i=1}^{\lambda}$, select the μ best: $D_\mu = \text{bestOf}_\mu(D)$
  - Compute the new mean $\hat{x}$ from D_μ

- This algorithm is called “(μ, λ)-ES” (Evolution Strategy)
  - The Gaussian is meant to represent a “species”

“Elitarian” Selection: $(\mu + \lambda)$-ES

- To make search monotonous(!), the algorithm also stores the previous elite D_μ
  - $\theta = (\hat{x}, D_\mu)$ now includes the mean $\hat{x}$ and previously selected

“Elitarian” Selection: $(\mu + \lambda)$-ES

- To make search monotonous(!), the algorithm also stores the previous elite D_μ
  - $\theta = (\hat{x}, D_\mu)$ now includes the mean $\hat{x}$ and previously selected

- The update heuristic $\theta \leftarrow h(\theta, D)$ selects from the union of new and elite:
  - Select the μ best $D_\mu \leftarrow \text{bestOf}_\mu(D_\mu \cup D)$
  - Compute the new mean $\hat{x}$ from D_μ

"Elitarian" Selection: $(\mu + \lambda)$-ES

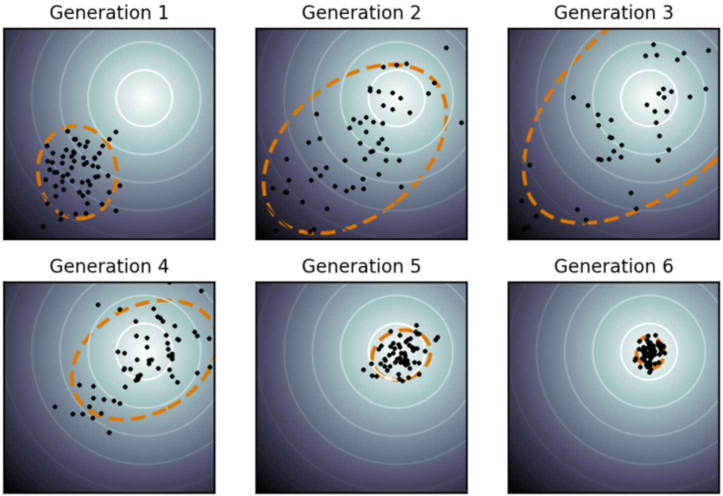
- To make search monotonous(!), the algorithm also stores the previous elite D_μ
  - $\theta = (\hat{x}, D_\mu)$ now includes the mean $\hat{x}$ and previously selected

- The update heuristic $\theta \leftarrow h(\theta, D)$ selects from the union of new and elite:
  - Select the μ best $D_\mu \leftarrow \text{bestOf}_\mu(D_\mu \cup D)$
  - Compute the new mean $\hat{x}$ from D_μ

- Special case: $(1 + 1)$**-ES = Greedy Local Search/Hill Climber**
- Special case: $(1 + \lambda)$**-ES = Local Search**

- Assuming a fixed σ and isotropic $\mathcal{N}(x \mid \hat{x}, \sigma^2)$ is limiting
  - No notion of going *forward* (downhill/momentum)
  - No adaptation of σ
  - Should steps smaller/larger/correlated depending on local Hessian!

Covariance Matrix Adaptation (CMA-ES)



[Hansen, N. (2006)]

Covariance Matrix Adaptation (CMA-ES)

- In Covariance Matrix Adaptation

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

  where C is the covariance matrix of the search distribution

- The θ maintains two more pieces of information: ϱ_σ and ϱ_c capture the "path" (motion) of the mean $\hat{x}$ in recent iterations
- Rough outline of the θ-update:
  - Let $D_\mu = \text{bestOf}_\mu(D)$ be the selected
  - Compute the new mean $\hat{x}$ of D_μ
  - Update ϱ_σ and ϱ_c proportional to $\hat{x}_{k+1} - \hat{x}_k$
  - Update σ depending on $|\varrho_\sigma|$
  - Update C depending on $\varrho_c \varrho_c^\top$ (rank-1-update) and $\text{Var}(D_\mu)$

CMA references

Hansen: *The CMA evolution strategy: a comparing review*. 2006

Hansen et al.: *Evaluating the CMA Evolution Strategy on Multimodal Test Functions*. PPSN 2004

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

CMA conclusions

- Good starting point for an off-the-shelf blackbox algorithm
- It includes components like estimating the local gradient ($\varrho_\sigma, \varrho_c$), the local "Hessian" (Var(D_μ)), smoothing out local minima (large populations)

- But is this tackling global optimization?

  "For "large enough" populations local minima are avoided"

  (But not really.)

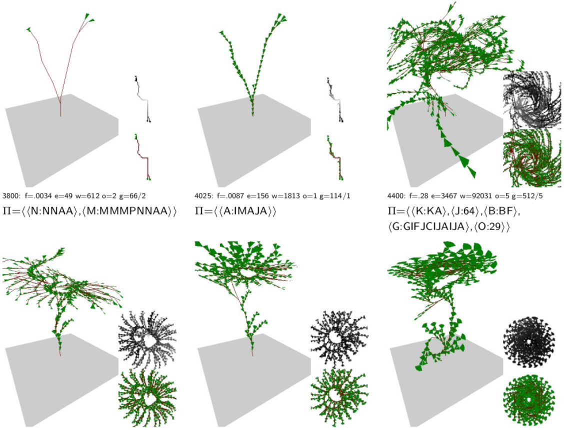
Estimation of Distribution Algorithms (EDAs)

- In general, θ can model a distribution $p_\theta(x)$ for any spaces (also discrete/hybrid) using any distribution representation (Bayesian Networks, probabilistic grammars, generative ML, etc)
- The update heuristic $\theta \leftarrow h(\theta, D)$ typically let's "$p_\theta(x)$ estimate D_μ", e.g. by likelihood maximization

$$\theta \leftarrow \underset{\theta}{\operatorname{argmin}} \; -\sum_{x \in D_\mu} \log p_\theta(x) + \text{regularization}$$

  - The regularization is important, otherwise the new offspring would "overfit" on the previous elite and not explore
  - E.g. ensure sufficient entropy

- Stochastic grammars to "learn" a distribution of selected structures



3800: f=.0034 e=49 w=612 o=2 g=66/2
$\Pi=\langle\langle N{:}NNAA\rangle,\langle M{:}MMMPNNAA\rangle\rangle$

4025: f=.0087 e=156 w=1813 o=1 g=114/1
$\Pi=\langle\langle A{:}IMAJA\rangle\rangle$

4400: f=.28 e=3467 w=92031 o=5 g=512/5
$\Pi=\langle\langle K{:}KA\rangle,\langle J{:}64\rangle,\langle B{:}BF\rangle,\langle G{:}GIFJCIJAIJA\rangle,\langle O{:}29\rangle\rangle$

5100: f=.20 e=1479 w=57134 o=2 g=217/2
$\Pi=\langle\langle K{:}\rangle,\langle J{:}77\rangle\rangle$

5400: f=.22 e=1410 w=59219 o=3 g=159/3
$\Pi=\langle\langle I{:}AJ\rangle,\langle C{:}JJCJ\rangle,\langle J{:}64\rangle\rangle$

8000: f=.31 e=3652 w=379288 o=2 g=141/2
$\Pi=\langle\langle I{:}39\rangle,\langle J{:}60\rangle\rangle$

[Toussaint, GECCO 2003]

Estimation of Distribution Algorithms (EDAs)

- EDAs *learn* correlations and structures in selected

  Agakov,..,Toussaint,..,: *Using Machine Learning to Focus Iterative Optimization*. CGO 2006

  Toussaint: *Compact representations as a search strategy: Compression EDAs*. Theoretical Computer Science, 2006

  - E.g., if in all selected distributions, the 3rd bit equals the 7th bit, then the search distribution $p_\theta(x)$ should put higher probability on such candidates
  - In discrete domains, graphical models can be used to learn the dependencies between variables, e.g. **Bayesian Optimization Algorithm (BOA)**
  - In continuous domains, CMA is an example for an EDA

Simulated Annealing (accepts also uphill steps)

- Could be viewed as extension to avoid getting stuck in local optima, which accepts steps with $f(y) > f(x)$ – but better viewed as sampling technique (see next page)

Input: initial point x ($\equiv \theta$), function $f(x)$, **proposal distribution** $q(y|x)$ ($\equiv p_x(y)$)

```
initialilze the temperature T = 1
repeat
    Sample single y ~ q(y|x)
    Acceptance probability A = min{1, e^{(f(x)-f(y))/T} q(x|y)/q(y|x)}
    With probability A update x <- y
    Decrease T, e.g. T <- (1 - ε)T for small ε
until x converges
```

Acceptance probability: $A = \min\left\{1,\ e^{\frac{f(x)-f(y)}{T}} \frac{q(x|y)}{q(y|x)}\right\}$

- Typically: $q(y|x) \propto \exp\{-\frac{1}{2}(y-x)^2/\sigma^2\}$
- Instance of our general scheme for $x \equiv \theta$, $p_\theta(x) \equiv q(x|\theta)$, $\lambda = 1$, update stochastic as above

Simulated Annealing

- Simulated Annealing is a Markov chain Monte Carlo (MCMC) method.
  - Must read!: *An Introduction to MCMC for Machine Learning*
  - These are iterative methods to sample from a distribution, in our case

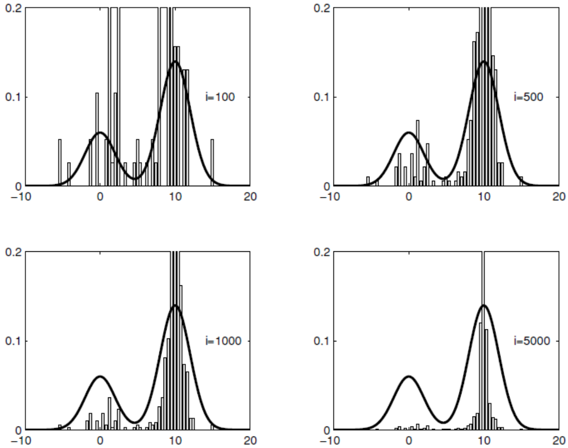
$$p(x) \propto e^{\frac{-f(x)}{T}}$$

- For a fixed temperature T, one can prove that the set of accepted points is distributed as $p(x)$ (but non-i.i.d.!) The acceptance probability

$$A = \min\left\{1, e^{\frac{f(x)-f(y)}{T}} \frac{q(x|y)}{q(y|x)}\right\}$$

  compares the $f(y)$ and $f(x)$, but also the reversibility of $q(y|x)$
- When cooling the temperature, samples focus at the extrema. Guaranteed to sample all extrema *eventually*

Simulated Annealing



[MCMC introduction (2003)]

Stochastic search conclusions

Input: initial θ, function $f(x)$, distribution model $p_\theta(x)$, update heuristic $h(\theta, D)$
Output: final θ and best point x

repeat
  Sample $\{x_i\}_{i=1}^{\lambda} \sim p_\theta(x)$
  Evaluate samples, $D = \{(x_i, f(x_i))\}_{i=1}^{\lambda}$
  Update $\theta \leftarrow h(\theta, D)$
until θ converges

- The framework is very general
- Algorithms differ in choice of θ, $p_\theta(x)$, and $h(t, D)$
- The update $h(\theta, D)$ "should train the distribution $p_\theta(x)$ to match good points"