

Estimation-of-Distribution Algorithms. Continuous Domain.

Petr Pošík

Dept. of Cybernetics
CTU FEE

Last week...	2
Intro to EDAs	3
Content of the lectures	4
Features of continuous spaces	5
The difference of binary and real space	6
Local neighborhood	7
Real-valued EDAs	8
Taxonomy	9
No Interactions Among Variables	10
Distribution Tree	11
Global Coordinate Transformations	12
Linear Coordinate Transformations	13
Mixture of Gaussians	14
Non-linear global transformation	15
Back to the Roots	16
Simple Gaussian EDA	17
Premature convergence	18
What happens on the slope?	19
Variance Enlargement in a Simple EDA	21
Summary of Continuous EDAs So Far	22
State of the Art	23
Current Trend	24
Preventing the Premature Convergence	25
AVS	26
AVS Triggers	27
AMS	28
Weighted ML Estimates	29
CMA-ES	30
Optimization via Classification	31
Remarks on SotA	32
Summary	33
Real-valued EDAs	34

Intro to EDAs

Black-box optimization

GA vs. EDA

- ✓ GA approach: select — *crossover* — *mutate*
- ✓ EDA approach: select — *model* — *sample*

EDA with binary representation

- ✓ the best possible (general, flexible) model: joint probability
 - ✗ determine the probability of each possible combination of bits
 - ✗ $2^D - 1$ parameters, exponential complexity
- ✓ less precise (less flexible), but simpler probabilistic models

Content of the lectures**Binary EDAs**

- ✓ Without interactions
 - ✗ 1-dimensional marginal probabilities $p(X = x)$
 - ✗ PBIL, UMDA, cGA
- ✓ Pairwise interactions
 - ✗ conditional probabilities $p(X = x|Y = y)$
 - ✗ sequences (MIMIC), trees (COMIT), forrest (BMDA)
- ✓ Multivariate interactions
 - ✗ conditional probabilities $p(X = x|Y = y, Z = z, \dots)$
 - ✗ Bayesian networks (BOA, EBNA, LFDA)

Continuous EDAs

- ✓ Histograms, mixtures of Gaussian distributions
- ✓ Analysis of a simple Gaussian EDA
- ✓ Remedies for premature convergence
 - ✗ Evolutionary strategies
 - ✗ AMS, Weighting, CMA-ES, classification

The difference of binary and real space

Binary space

- ✓ Each possible solution is placed in one of the corners of D -dimensional hypercube
- ✓ No values lying between them
- ✓ Finite number of elements
- ✓ Not possible to make 2 or more steps in the same *direction*

Real space

- ✓ The space in each dimension need not be bounded
- ✓ Even when bounded by a hypercube, there are infinitely many points between the bounds (theoretically; in practice we are limited by the numerical precision of given machine)
- ✓ Infinitely many (even uncountably many) candidate solutions

Local neighborhood

How do you define a local neighborhood?

- ✓ ... as a set of points that do not have the distance to a reference point larger than a threshold?
 - ✗ The volume of the local neighborhood relative to the volume of the whole space exponentially drops
 - ✗ With increasing dimensionality the neighborhood becomes increasingly more local
- ✓ ... as a set of points that are closest to the reference point and their unification covers part of the search space of certain (constant) size?
 - ✗ The size of the local neighborhood rises with dimensionality of the search space
 - ✗ With increasing dimensionality of the search space the neighborhood is increasingly less local

Another manifestation of the **curse of dimensionality!**

Taxonomy

2 basic approaches:

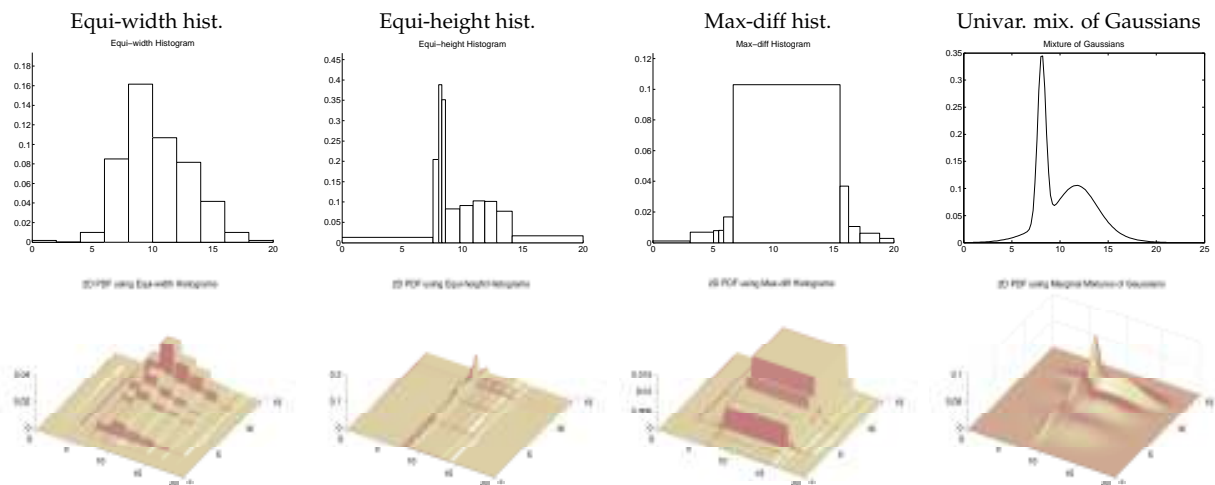
- ✓ discretize the representation and use EDA with discrete model
- ✓ use EDA with natively continuous model

Again, classification based on the interactions complexity they can handle:

- ✓ Without interactions
 - ✗ UMDA: model is product of univariate marginal models, only their type is different
 - ✗ Univariate histograms?
 - ✗ Univariate Gaussian distribution?
 - ✗ Univariate mixture of Gaussians?
- ✓ Pairwise and higher-order interactions:
 - ✗ Many different types of interactions!
 - ✗ Model which would describe all possible kinds of interaction is virtually impossible to find!

No Interactions Among Variables

UMDA: EDA with marginal product model $p(\mathbf{x}) = \prod_{d=1}^D p(x_d)$

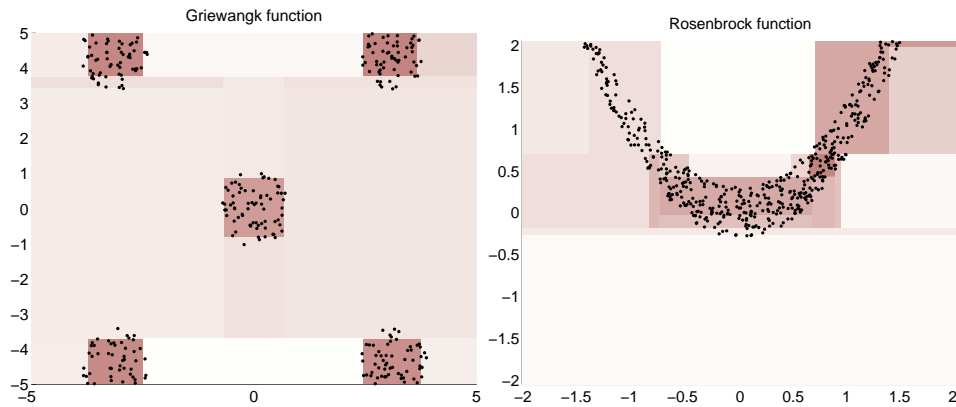


Lessons learned:

- ✓ If a separable function is rotated, UMDA does not work.
- ✓ If there are nonlinear interactions, UMDA does not work.
- ✓ EDAs with univariate marginal product models are not flexible enough!
- ✓ We need EDAs that can handle some kind of interactions!

Distribution Tree

Distribution Tree-Building Real-valued EA [Poš04]



Distribution-Tree model

- ✓ identifies hyper-rectangular areas of the search space with significantly different densities
- ✓ can handle certain type of interactions

Lessons learned:

- ✓ Cannot model promising areas not aligned with the coordinate axes.
- ✓ *We need models able to rotate the coordinate system!*

[Poš04] Petr Pošík. Distribution tree-building real-valued evolutionary algorithm. In *Parallel Problem Solving From Nature — PPSN VIII*, pages 372–381, Berlin, 2004. Springer. ISBN 3-540-23092-0.

P. Pošík © 2011

A0M33EOA: Evolutionary Optimization Algorithms – 11 / 34

Global Coordinate Transformations

Algorithm 1: EDA with global coordinate transformation

```
1 begin
2   Initialize the population.
3   while termination criteria are not met do
4     Select parents from the population.
5     Transform the parents to a space where the variables are independent of each other.
6     Learn a model of the transformed parents distribution.
7     Sample new individuals in the transformed space.
8     Transform the offspring back to the original space.
9     Incorporate offspring into the population.
```

The individuals are

- ✓ evaluated in the original space (where the fitness function is defined), but
- ✓ bred in the transformed space (where the dependencies are reduced).

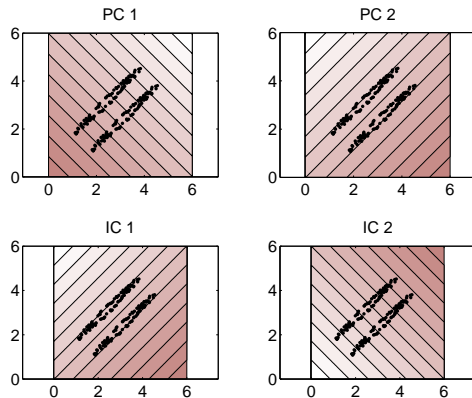
P. Pošík © 2011

A0M33EOA: Evolutionary Optimization Algorithms – 12 / 34

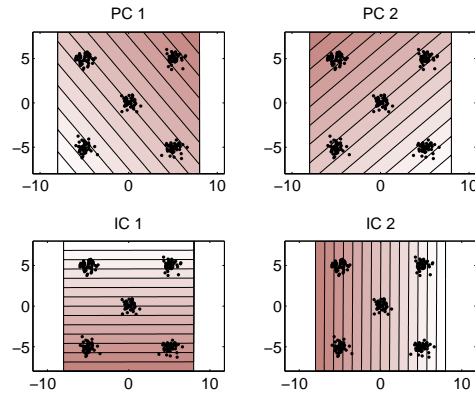
Linear Coordinate Transformations

UMDA with equi-height histogram models [Poš05]:

- ✓ No transformation vs. PCA vs. ICA
- ✓ PCA and ICA are used to find a suitable rotation of the space, not to reduce the space dimensionality



Different results: the difference does not matter.



Different results: the difference matters!

Lessons learned:

- ✓ The global information extracted by linear transformations was often not useful.
- ✓ We need non-linear transformations or local transformations!!!

[Poš05] Petr Pošík. On the utility of linear transformations for population-based optimization algorithms. In *Preprints of the 16th World Congress of the International Federation of Automatic Control, Prague, 2005*. IFAC. CD-ROM.

Mixture of Gaussians

Gaussian mixture model (GMM):

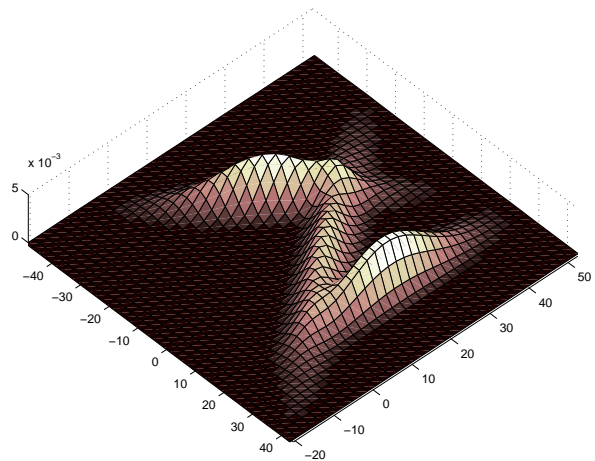
$$P(x) = \sum_{k=1}^K \alpha_k \mathcal{N}(x | \mu_k, \Sigma_k) \quad (1)$$

Normalization and the requirement of positivity:

$$\sum_{k=1}^K \alpha_k = 1$$

$$0 \leq \alpha_k \leq 1$$

Model learned by EM algorithm.

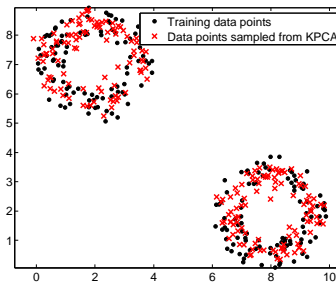


Lessons learned:

- ✓ GMM is able to model locally linear dependencies.
- ✓ We need to specify the number of components beforehand!
- ✓ If the optimum is not covered by at least one of the Gaussian peaks, the EA will miss it!

Non-linear global transformation

Kernel PCA as the transformation technique in EDA [Poš04]



Works too well:

- ✓ It reproduces the pattern with high fidelity
- ✓ If the population is not centered around the optimum, the EA will miss it

Lessons learned:

- ✓ *Continuous EDA must be able to effectively move the whole population!!!*
- ✓ *Is the MLE principle actually suitable for model building in EAs???*

[Poš04] Petr Pošík. Using kernel principal components analysis in evolutionary algorithms as an efficient multi-parent crossover operator. In *IEEE 4th International Conference on Intelligent Systems Design and Applications*, pages 25–30, Piscataway, 2004. IEEE. ISBN 963-7154-29-9.

Back to the Roots

Simple Gaussian EDA

Consider a simple EDA with the following settings:

Algorithm 2: Gaussian EDA

```

1 begin
2   {μ1, Σ1} ← InitializeModel()
3   g ← 1
4   while not TerminationCondition() do
5     X ← SampleGaussian(μg, k · Σg)
6     f ← Evaluate(X)
7     Xsel ← Select(X, f, τ)
8     {μg+1, Σg+1} ← LearnGaussian(Xsel)
9     g ← g + 1

```

- ✓ **Generational model:** no member of the current population survives to the next one
- ✓ **Truncation selection:** use $\tau \cdot N$ best individuals to build the model
- ✓ **Gaussian distribution:** fit the Gaussian using maximum likelihood (ML) estimate

Gaussian distribution:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\}$$

Maximum likelihood (ML) estimates of parameters

$$\boldsymbol{\mu}_{\text{ML}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n, \text{ where } \mathbf{x}_n \in \mathbf{X}_{\text{sel}}$$

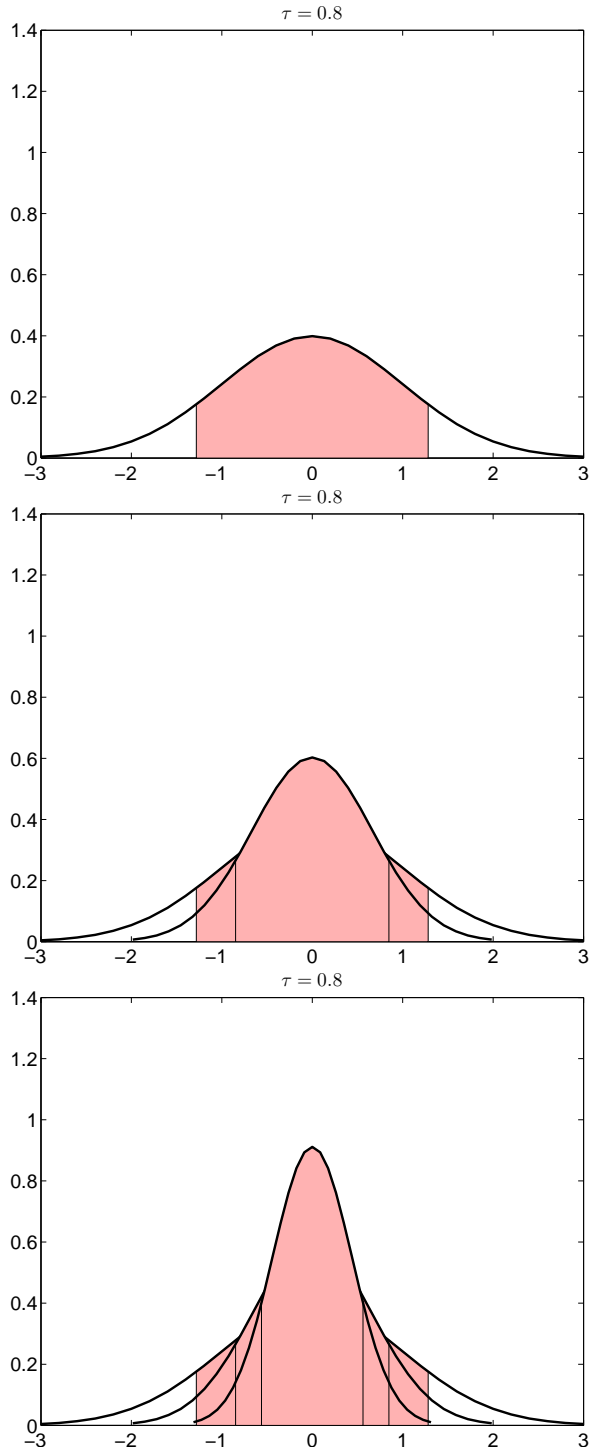
$$\boldsymbol{\Sigma}_{\text{ML}} = \frac{1}{N-1} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}_{\text{ML}})(\mathbf{x}_n - \boldsymbol{\mu}_{\text{ML}})^T$$

Premature convergence

Using Gaussian distribution and ML estimation seems as a good idea...
 ... but it is actually very bad optimizer!!!

Two situations:

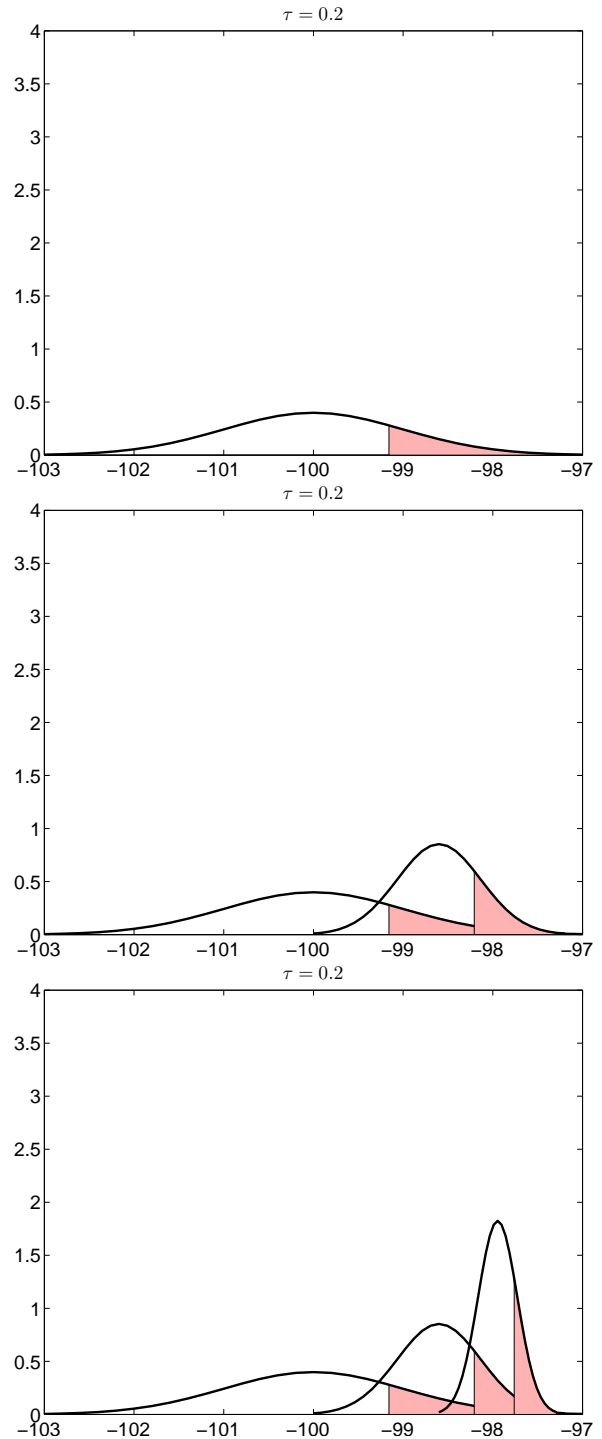
Population centered around optimum (population in the valley):



Algorithm works:

- ✓ the optimum is located
- ✓ the algorithm *focuses* the population on the optimum

Population far away from optimum (population on the slope):



Algorithm fails:

- ✓ the optimum is far away
- ✓ the algorithm is not able to *shift* the population towards optimum

What happens on the slope?

The change of population statistics in 1 generation:

Expected value:

$$\mu^{t+1} = E(X|X > x_{\min}) = \mu^t + \sigma^t \cdot d(\tau),$$

Variance:

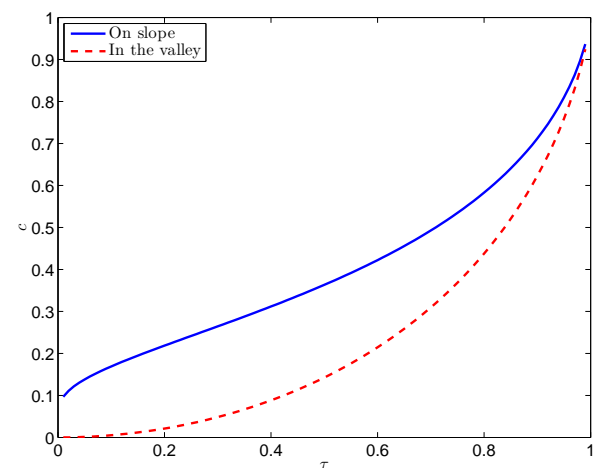
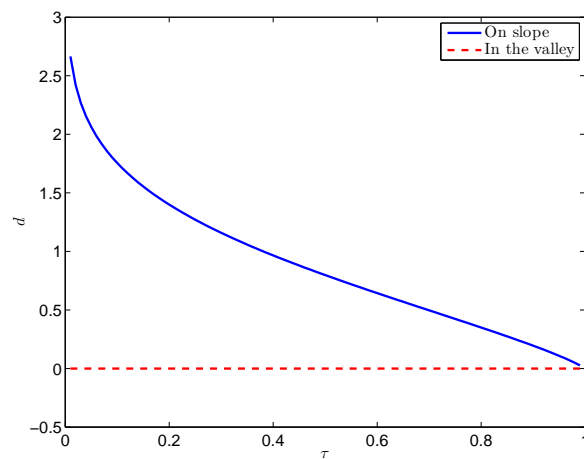
$$(\sigma^{t+1})^2 = \text{Var}(X|X > x_{\min}) = (\sigma^t)^2 \cdot c(\tau),$$

where

$$d(\tau) = \frac{\phi(\Phi^{-1}(\tau))}{\tau}.$$

where

$$c(\tau) = 1 + \frac{\Phi^{-1}(1-\tau) \cdot \phi(\Phi^{-1}(\tau))}{\tau} - d(\tau)^2.$$



What happens on the slope (cont.)

Population statistics in generation t :

$$\mu^t = \mu^0 + \sigma^0 \cdot d(\tau) \cdot \sum_{i=1}^t \sqrt{c(\tau)^{i-1}}$$

$$\sigma^t = \sigma^0 \cdot \sqrt{c(\tau)^t}$$

Convergence of population statistics:

$$\lim_{t \rightarrow \infty} \mu^t = \mu^0 + \sigma^0 \cdot d(\tau) \cdot \frac{1}{1 - \sqrt{c(\tau)}}$$

$$\lim_{t \rightarrow \infty} \sigma^t = 0$$

Geometric series

The distance the population can “travel” in this algorithm is **bounded!**

Premature convergence!

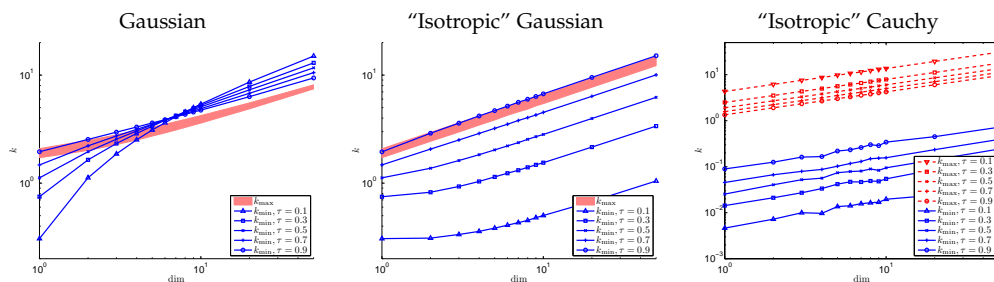
Lessons learned:

- ✓ Maximum likelihood estimates are suitable in situations when model fits the fitness function well (at least in local neighborhood)
 - ✗ Gaussian distribution may be suitable in the neighborhood of optimum.
 - ✗ Gaussian distribution is not suitable on the slope of fitness function!
- ✓ We need something different from MLE to traverse the slopes!!!

Variance Enlargement in a Simple EDA

What happens if we enlarged the MLE estimate of variance with a constant multiplier k ? [Poš08]

- ✓ What is the minimal value k_{\min} ensuring that the model will not converge on the slope?
- ✓ What is the maximal value k_{\max} ensuring that the model will not diverge in the valley?
- ✓ Is there a single value k of the multiplier for MLE variance estimate that would ensure a reasonable behavior in both situations?
- ✓ Does it depend on the type of the single-peak distribution being used?



- ✓ For Gaussian and “isotropic Gaussian”, allowable k is hard or impossible to find.
- ✓ For isotropic Cauchy, allowable k seems to always exist...
 - ✗ ...but this does not guarantee a reasonable behavior.

[Poš08] Petr Pošík. Preventing premature convergence in a simple EDA via global step size setting. In Günther Rudolph, editor, *Parallel Problem Solving from Nature – PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 549–558. Springer, 2008.

Summary of Continuous EDAs So Far

Initially, high expectations:

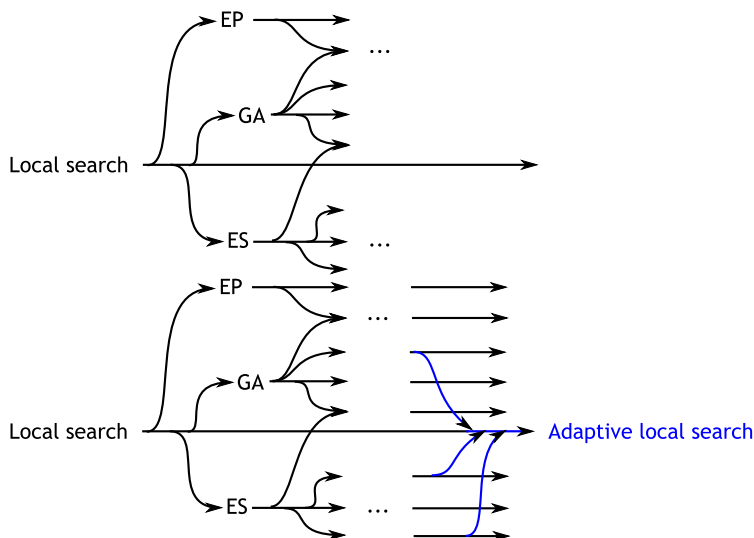
- ✓ Started with structurally simple models for complex objective functions.
 - ✗ They did not work, partially because of the discrepancy between the complexities of the model and the function.
- ✓ Used increasingly complex and flexible models.
 - ✗ Some improvements were gained, but even the most complex models did not fulfill the expectations.
- ✓ Realized that a fundamental mistake was present all the time:
 - ✗ MLE principle builds models which try to reconstruct the points they were build upon.
 - ✗ This allows to focus on already covered areas, but not to shift the population to unexplored places.

Current research directions:

- ✓ Aimed at understanding and developing principles critical for successful continuous EDAs.
 - ✗ Studying behavior on simple functions first.
 - ✗ Using simple, single-peak models so that the resulting algorithm behave (more or less) as local search procedures.

State of the Art

Current Trend: Population-based Adaptive Local Search



There's something about the population:

- ✓ data set forming a basis for offspring creation
- ✓ allows for searching the space in several places at once
- ✓ ~~allows for searching the space in several places at once~~
(replaced by restarted local search with adaptive neighborhood)

Hypothesis:

- ✓ The data set (population) is very useful when creating (sometimes implicit) global model of the fitness landscape or a local model of the neighborhood.
- ✓ It is often better to have a robust adaptive local search procedure and restart it, than to deal with a complex global search algorithm.

Preventing the Premature Convergence

- ✓ self-adaptation of the variance [OKHK04] (let the variance be part of the chromosome)
- ✓ adaptive variance scaling when population is on the slope, ML estimate of variance when population is in the valley
- ✓ anticipate the shift of the mean and move part of the offspring in the anticipated direction
- ✓ use weighted estimates of distribution parameters
- ✓ do not estimate the distribution of selected points, but rather a distribution of selected mutation steps
- ✓ use a different principle to estimate the parameters of the Gaussian

[OKHK04] Jiří Očenášek, Stefan Kern, Nikolaus Hansen, and Petros Koumoutsakos. A mixed bayesian optimization algorithm with variance adaptation. In Xin Yao, editor, *Parallel Problem Solving from Nature – PPSN VIII*, pages 352–361. Springer-Verlag, Berlin, 2004.

Adaptive Variance Scaling

AVS [GBR06]:

- ✓ Enlarge the ML estimate of Σ by an *adaptive* coefficient c_{AVS}
- ✓ If an improvement was not found in the current generation, we explore to much, thus decrease c_{AVS} : $c_{AVS} \leftarrow \eta^{DEC} c_{AVS}$, $\eta^{DEC} \in (0, 1)$
- ✓ If an improvement was found in the current generation, we may get better results with increased c_{AVS} : $c_{AVS} \leftarrow \eta^{INC} c_{AVS}$, $\eta^{INC} > 1$
- ✓ c_{AVS} is bounded: $1 \leq c_{AVS} \leq c^{AVS-MIN}$

[GBR06] Jörn Grahl, Peter A. N. Bosman, and Franz Rothlauf. The correlation-triggered adaptive variance scaling IDEA. In *Proceedings of the 8th annual conference on Genetic and Evolutionary Computation Conference – GECCO 2006*, pages 397–404, New York, NY, USA, 2006. ACM Press.

AVS Triggers

With AVS, all improvements increase c_{AVS} :

- ✓ This is not always needed, especially in the valleys.
- ✓ Trigger AVS when on slope; in the valley, use ordinary MLE.

Correlation trigger for AVS (CT-AVS) [GBR06]:

- ✓ Compute the ranked correlation coefficient of p.d.f. values and function values, $p(\mathbf{x}_i)$ and $f(\mathbf{x}_i)$.
- ✓ If the distribution is placed around optimum, function values increase with decreasing p.d.f., correlation will be large. Use ordinary MLE.
- ✓ If the distribution is on a slope, correlation will be close to zero. Use AVS.

Standard-deviation ratio trigger for AVS (SDR-AVS) [BGR07]:

- ✓ Compute \bar{x}^{IMP} as the average of all improving individuals in the current population
- ✓ If $p(\bar{x}^{\text{IMP}})$ is "low" (the improvements are found far away from the distribution center), we are probably on a slope. Use AVS.
- ✓ If $p(\bar{x}^{\text{IMP}})$ is "high" (the improvements are found near the distribution center), we are probably in a valley. Use ordinary MLE.

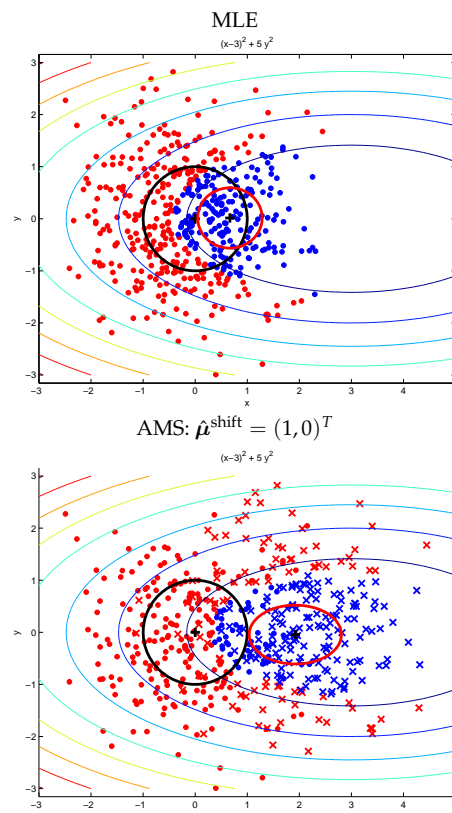
[BGR07] Peter A. N. Bosman, Jörn Grahl, and Franz Rothlauf. SDR: A better trigger for adaptive variance scaling in normal EDAs. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and Evolutionary Computation*, pages 492–499, New York, NY, USA, 2007. ACM Press.

[GBR06] Jörn Grahl, Peter A. N. Bosman, and Franz Rothlauf. The correlation-triggered adaptive variance scaling IDEA. In *Proceedings of the 8th annual conference on Genetic and Evolutionary Computation Conference – GECCO 2006*, pages 397–404, New York, NY, USA, 2006. ACM Press.

Anticipated Mean Shift

Anticipated mean shift (AMS) [BGT08]:

- ✓ AMS is defined as: $\hat{\boldsymbol{\mu}}^{\text{shift}} = \hat{\boldsymbol{\mu}}(t) - \hat{\boldsymbol{\mu}}(t-1)$
- ✓ AMS is an estimate of the direction of improvement
- ✓ 100% of offspring are moved by certain fraction of AMS:
 $\mathbf{x} = \mathbf{x} + \delta \hat{\boldsymbol{\mu}}^{\text{shift}}$
- ✓ When centered around optimum, $\hat{\boldsymbol{\mu}}^{\text{shift}} = 0$ and the original approach is unchanged.
- ✓ Selection must choose parent from both the old and the shifted regions to adjust Σ suitably.

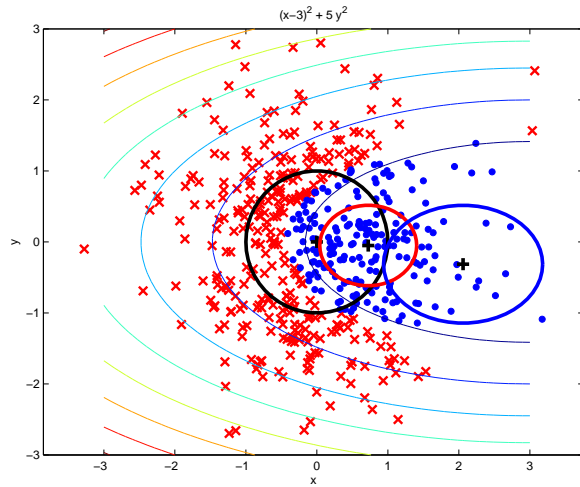


[BGT08] Peter Bosman, Jörn Grahl, and Dirk Thierens. Enhancing the performance of maximum-likelihood Gaussian EDAs using anticipated mean shift. In Günter Rudolph et al., editor, *Parallel Problem Solving from Nature – PPSN X*, volume 5199 of LNCS, pages 133–143. Springer, 2008.

Weighted ML Estimates

Account for the values of p.d.f. of the selected parents \mathbf{X}_{sel} [TT09]:

- ✓ assign weights inversely proportional to the values of p.d.f.



[TT09] Fabien Teytaud and Olivier Teytaud. Why one must use reweighting in estimation of distribution algorithms. In *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 453–460, New York, NY, USA, 2009. ACM.

Weighted (ML) estimates of parameters

$$\boldsymbol{\mu}_W = \frac{1}{V_1} \sum_{i=1}^N w_i \mathbf{x}_i, \text{ where } \mathbf{x}_i \in \mathbf{X}_{\text{sel}}$$

$$\boldsymbol{\Sigma}_W = \frac{V_1}{V_1^2 - V_2} \sum_{i=1}^N w_i (\mathbf{x}_i - \boldsymbol{\mu}_{\text{ML}})(\mathbf{x}_i - \boldsymbol{\mu}_{\text{ML}})^T$$

where

$$w_i = \frac{1}{p(\mathbf{x}_i)}$$

$$V_1 = \sum w_i$$

$$V_2 = \sum w_i^2$$

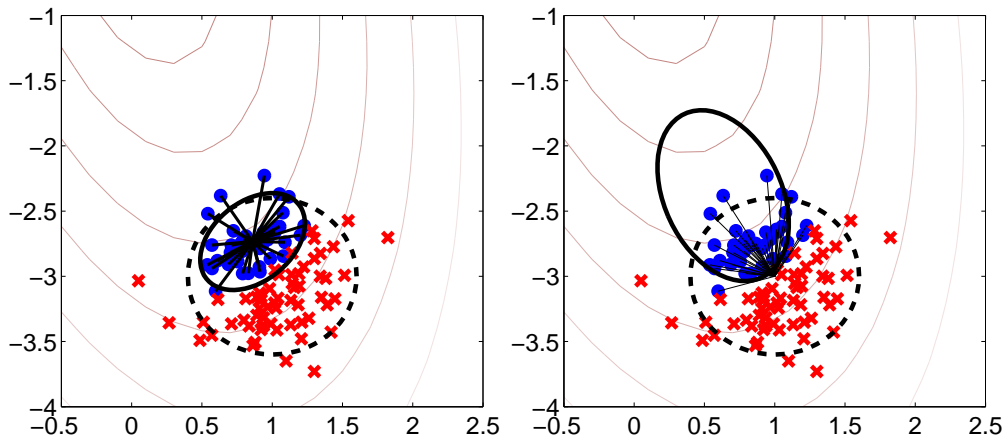
CMA-ES

Evolutionary strategy with cov. matrix adaptation [HO01]

- ✓ $(\mu/\mu, \lambda)$ -ES (recombinative, mean-centric)
- ✓ model is adapted, not built from scratch each generation
- ✓ accumulates the successful steps over many generations

Compare:

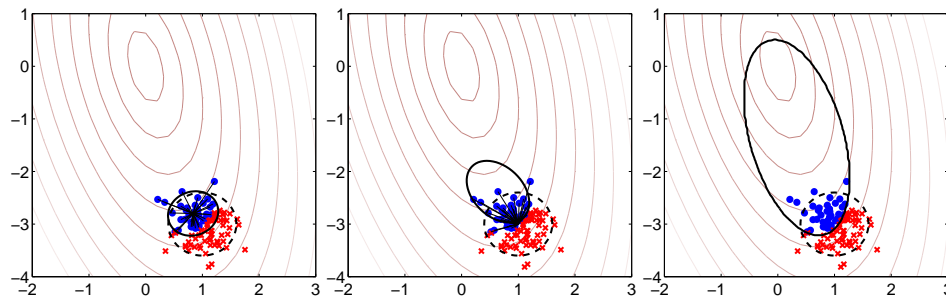
- ✓ Simple Gaussian EDA estimates the distribution of selected individuals (left fig.)
- ✓ CMA-ES estimates the distribution of successful mutation steps (right fig.)



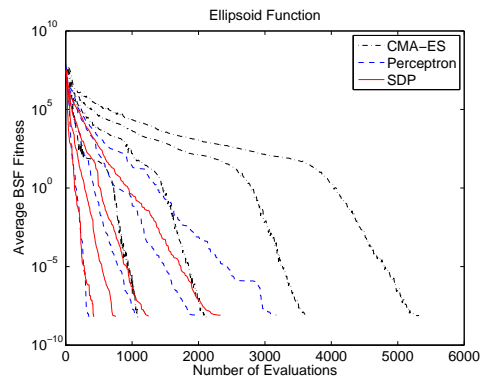
[HO01] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

Optimization via Classification

Build a quadratic classifier separating the selected and the discarded individuals [PF07]



- ✓ Classifier built by modified perceptron algorithm or by semidefinite programming
- ✓ Works well for pure quadratic functions
- ✓ If the selected and discarded individuals are not separable by an ellipsoid, the training procedure fails to create a good model
- ✓ Work in progress; not solved yet



[PF07] Petr Pošík and Vojtěch Franc. Estimation of fitness landscape contours in EAs. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 562–569, New York, NY, USA, 2007. ACM Press.

Remarks on SotA

- ✓ Many techniques to fight premature convergence
- ✓ Although based on different principles, some of them converge to similar algorithms (weighted MLE, CMA-ES, NES)
- ✓ Only a few sound principles; the most of them are heuristic approaches

Real-valued EDAs

- ✓ much less developed than EDAs for binary representation
- ✓ the difficulties are caused mainly by
 - ✗ much more severe effects of the curse of dimensionality
 - ✗ many different types of interactions among variables
- ✓ Gaussian distribution used most often, but pure maximum-likelihood estimates are BAD! Some other remedies are needed.
- ✓ Despite of that, EDA (and EAs generally) are able to gain better results then conventional optimization techniques (line search, Nelder-Mead search, ...)