Evolution Strategies

Basic Introduction

Prof. Dr. Thomas Bäck
NuTech Solutions, Inc., Charlotte, NC
baeck@nutechsolutions.de
and
Leiden University, The Netherlands
baeck@liacs.nl

1

ACM 978-1-60558-131-6/08/07.

Abstract

This tutorial gives a basic introduction to **evolution strategies**, a class of evolutionary algorithms. Key features such as mutation, recombination and selection operators are explained, and specifically the concept of **self-adaptation** of strategy parameters is introduced.

All algorithmic concepts are explained to a level of detail such that an implementation of basic evolution strategies is possible.

Some guidelines for utilization as well as some application examples are given.

2

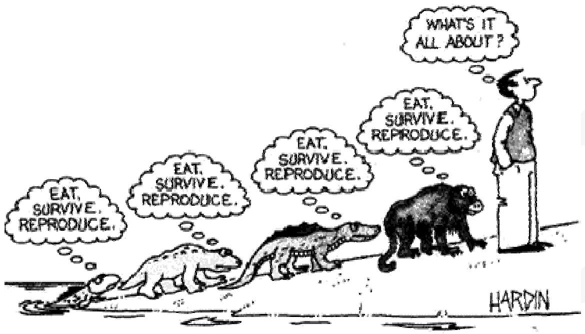
Overview

- Introduction: Optimization and EAs
- Evolution Strategies
- Further Developments
  - Multi-Criteria Optimization
  - Mixed-Integer ES
  - Soft-Constraints
  - Response Surface Approximation
- Examples

3

Background I

- Biology = Engineering (Daniel Dennett)



4

Introduction

- Modeling
  - !!! ⇒ ??? ⇒ !!!
  - Input: Known (measured)
  - Output: Known (measured)
  - Interrelation: Unknown
- Simulation
  - !!! ⇒ !!! ⇒ ???
  - Input: Will be given
  - How is the result for the input?
  - Model: Already exists
- Optimization
  - ??? ⇒ !!! ⇒ !!!
  - Objective: Will be given
  - How (with which parameter settings) to achieve this objective?

Universiteit Leiden

5

Simulation vs. Optimization

… what happens if?

Simulator → Result

Trial & Error

... how do I achieve the best result?

Optimizer → Simulator → Optimal Result

Maximization / Minimization
If so, multiple objectives

Universiteit Leiden

6

Introduction:

Optimization
Evolutionary Algorithms

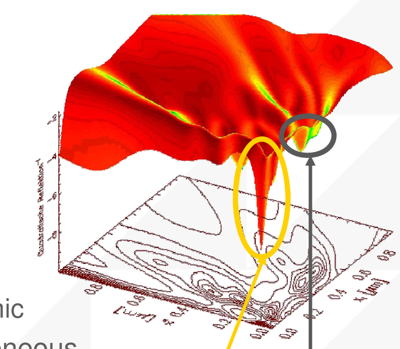
Universiteit Leiden

7

Optimization

$f : M \rightarrow \Re, f(\vec{x}) \rightarrow \min$

- f : objective function
  - High-dimensional
  - Non-linear, multimodal
  - Discontinuous, noisy, dynamic
- $M \subseteq M_1 \times M_2 \times \ldots \times M_n$ heterogeneous
- Restrictions possible over $M, f(\vec{x})$
- Good local, robust optimum desired
- Realistic landscapes are like that!



Universiteit Leiden

8

Optimization Creating Innovation

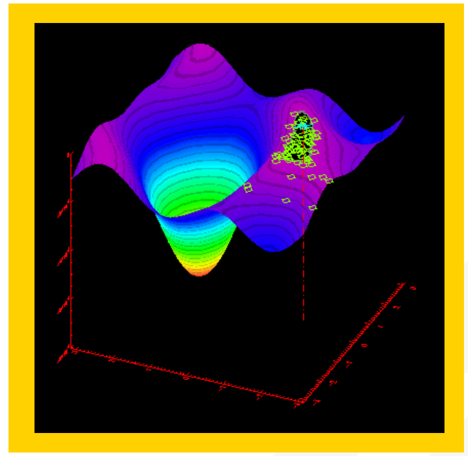
- Illustrative Example: Optimize Efficiency
  - Initial:

  - Evolution:

- 32% Improvement in Efficiency !

9

Dynamic Optimization



- Dynamic Function
- 30-dimensional
- 3D-Projection

10

Classification of Optimization Algorithms

- Direct optimization algorithm:
  Evolutionary Algorithms

$f(\vec{x})$

- First order optimization algorithm:
  e.g., gradient method

$f(\vec{x}), \nabla f(\vec{x})$

- Second order optimization algorithm:
  e.g., Newton method

$f(\vec{x}), \nabla f(\vec{x}), \nabla^2 f(\vec{x})$

11

Iterative Optimization Methods

- General description:

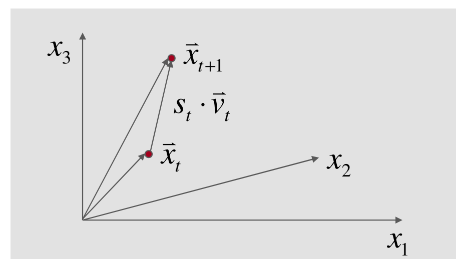
New Point

Actual Point

$$\vec{x}_{t+1} = \vec{x}_t + s_t \cdot \vec{v}_t$$

Directional vector

Step size (scalar)



- At every Iteration:
  - Choose direction
  - Determine step size
- Direction:
  - Gradient
  - Random
- Step size:
  - 1-dim. optimization
  - Random
  - Self-adaptive

12

The Fundamental Challenge

- Global convergence with probability one:

$$\lim_{t\to\infty} \Pr(\vec{x}^* \in P(t)) = 1$$

- General, but for practical purposes useless
- Convergence velocity:

$$\varphi = E(f_{\max}(P(t+1)) - f_{\max}(P(t)))$$

- Local analysis only, specific (convex) functions

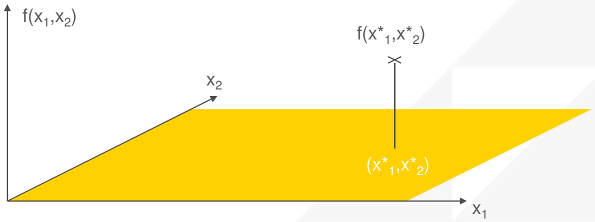
Theoretical Statements

- Global convergence (with probability 1):

$$\lim_{t\to\infty} \Pr(\vec{x}^* \in P(t)) = 1$$

- General statement (holds for all functions)
- Useless for practical situations:
  - Time plays a major role in practice
  - Not all objective functions are relevant in practice

An Infinite Number of Pathological Cases !



- NFL-Theorem:
  - All optimization algorithms perform equally well iff performance is averaged over all possible optimization problems.
- Fortunately: We are not Interested in „all possible problems“

Theoretical Statements

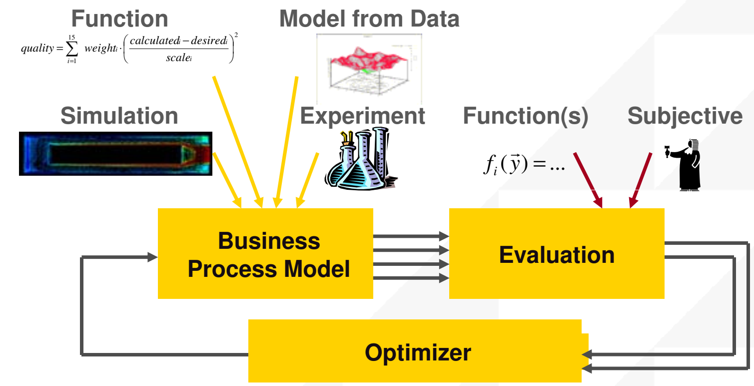
- Convergence velocity:

$$\varphi = E(f_{\max}(P(t+1)) - f_{\max}(P(t)))$$

- Very specific statements
  - Convex objective functions
  - Describes convergence in local optima
  - Very extensive analysis for Evolution Strategies

Evolution Strategies

Model-Optimization-Action



Evolutionary Algorithms Taxonomy

Evolutionary Algorithms

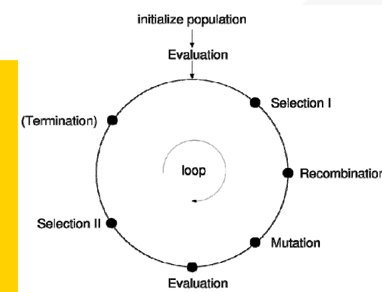
Evolution Strategies
- Mixed-integer capabilities
- Emphasis on mutation
- Self-adaptation
- Small population sizes
- Deterministic selection
- Developed in Germany
- Theory focused on convergence velocity

Genetic Algorithms
- Discrete representations
- Emphasis on crossover
- Constant parameters
- Larger population sizes
- Probabilistic selection
- Developed in USA
- Theory focused on schema processing

Other
- Evolutionary Progr.
- Differential Evol.
- GP
- PSO
- EDA
- Real-coded Gas
- …

Generalized Evolutionary Algorithm

t := 0;
***initialize*(P(t));**
***evaluate*(P(t));**
while not *terminate* do
    P'(t) := *mating_selection*(P(t));
    P''(t) := *variation*(P'(t));
    ***evaluate*(P''(t));**
    P(t+1) := *environmental_selection*(P''(t) ∪ Q);
    t := t+1;
od



Evolution Strategy – Basics

- Mostly real-valued search space IR^n
  - also mixed-integer, discrete spaces
- Emphasis on mutation
  - n-dimensional normal distribution
  - expectation zero
- Different recombination operators
- Deterministic selection
  - (μ, λ)-selection: Deterioration possible
  - $(\mu+\lambda)$-selection: Only accepts improvements
- $\lambda >> \mu$, i.e.: Creation of offspring surplus
- Self-adaptation of strategy parameters.

NuTech Solutions — Universiteit Leiden

21

Representation of search points

- Simple ES with 1/5 success rule:
  - Exogenous adaptation of step size σ
  - Mutation: $N(0, \sigma)$

$$\vec{a} = (x_1, \ldots, x_n)$$

- Self-adaptive ES with single step size:
  - One σ controls mutation for all x_i
  - Mutation: $N(0, \sigma)$

$$\vec{a} = ((x_1, \ldots, x_n), \sigma)$$

NuTech Solutions — Universiteit Leiden

22

Representation of search points

- Self-adaptive ES with individual step sizes:
  - One individual σ_i per x_i
  - Mutation: $N_i(0, \sigma_i)$

$$\vec{a} = ((x_1, \ldots, x_n), (\sigma_1, \ldots, \sigma_n))$$

- Self-adaptive ES with correlated mutation:
  - Individual step sizes
  - One correlation angle per coordinate pair
  - Mutation according to covariance matrix: $N(\mathbf{0}, \mathbf{C})$

$$\vec{a} = ((x_1, \ldots, x_n), (\sigma_1, \ldots, \sigma_n), (\alpha_1, \ldots, \alpha_{n(n-1)/2}))$$

NuTech Solutions — Universiteit Leiden

23

Evolution Strategy:

Algorithms
Mutation

NuTech Solutions — Universiteit Leiden

24

Operators: Mutation – one σ

- Self-adaptive ES with one step size:
  - One σ controls mutation for all x_i
  - Mutation: N(0, σ)

$$\vec{a} = ((x_1,\ldots,x_n),\sigma)$$
$$\vec{a}' = ((x_1',\ldots,x_n'),\sigma')$$
$$\sigma' = \sigma \cdot \exp(\tau_0 \cdot N(0,1))$$
$$x_i' = x_i + \sigma' \cdot N_i(0,1)$$

Individual before mutation

Individual after mutation

1.: Mutation of step sizes

2.: Mutation of objective variables

Here the new σ' is used!

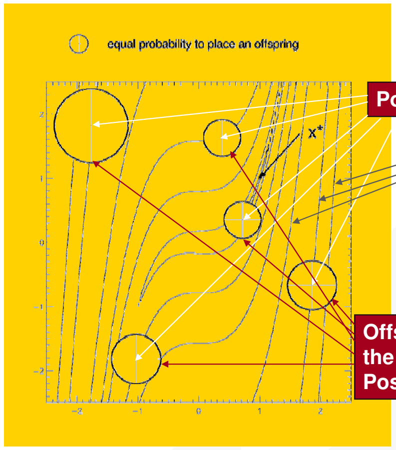
Operators: Mutation – one σ

- Thereby τ_0 is the so-called learning rate
  - Affects the speed of the σ-Adaptation
  - τ_0 bigger: faster but more imprecise
  - τ_0 smaller: slower but more precise
  - How to choose τ_0?
  - According to recommendation of Schwefel*:

$$\tau_0 = \frac{1}{\sqrt{n}}$$

*H.-P. Schwefel: Evolution and Optimum Seeking, Wiley, NY, 1995.

Operators: Mutation – one σ



Position of parents (here: 5)

Contour lines of objective function

Offspring of parent lies on the hyper sphere (for n > 10); Position is uniformly distributed

Pros and Cons: One σ

- Advantages:
  - Simple adaptation mechanism
  - Self-adaptation usually fast and precise
- Disadvantages:
  - Bad adaptation in case of complicated contour lines
  - Bad adaptation in case of very differently scaled object variables
    - -100 < x_i < 100 and e.g. -1 < x_j < 1

Operators: Mutation – individual σ_i

- Self-adaptive ES with individual step sizes:
  - One σ_i per x_i
  - Mutation: $N_i(0, \sigma_i)$

$$\vec{a} = ((x_1,\ldots,x_n),(\sigma_1,\ldots,\sigma_n))$$
$$\vec{a}' = ((x_1',\ldots,x_n'),(\sigma_1',\ldots,\sigma_n'))$$
$$\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1))$$
$$x_i' = x_i + \sigma_i' \cdot N_i(0,1)$$

Individual before Mutation

Individual after Mutation

1.: Mutation of individual step sizes

2.: Mutation of object variables

The new individual σ_i' are used here!

Operators: Mutation – individual σ_i

- τ, τ' are learning rates, again
  - τ': Global learning rate
  - N(0,1): Only one realisation
  - τ: local learning rate
  - $N_i(0,1)$: n realisations
  - Suggested by Schwefel*:

$$\tau' = \frac{1}{\sqrt{2n}} \qquad \tau = \frac{1}{\sqrt{2\sqrt{n}}}$$

*H.-P. Schwefel: Evolution and Optimum Seeking, Wiley, NY, 1995.

Operators: Mutation – individual σ_i

Position of parents (here: 5)

Contour lines

Offspring are located on the hyperellipsoid (für n > 10); position equally distributed.

Pros and Cons: Individual σ_i

- Advantages:
  - Individual scaling of object variables
  - Increased global convergence reliability
- Disadvantages:
  - Slower convergence due to increased learning effort
  - No rotation of coordinate system possible
    - Required for badly conditioned objective function

Operators: Correlated Mutations

- Self-adaptive ES with correlated mutations:
  - Individual step sizes
  - One rotation angle for each pair of coordinates
  - Mutation according to covariance matrix: N(**0**, **C**)

$$\vec{a} = ((x_1,\ldots,x_n),(\sigma_1,\ldots,\sigma_n),(\alpha_1,\ldots,\alpha_{n(n-1)/2}))$$ Individual before

$$\vec{a}' = ((x_1',\ldots,x_n'),(\sigma_1',\ldots,\sigma_n'),(\alpha_1',\ldots,\alpha_{n(n-1)/2}'))$$ Individual after mutation

$$\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1))$$ 1.: Mutation of Individual step sizes

$$\alpha_j' = \alpha_j + \beta \cdot N_j(0,1)$$ 2.: Mutation of rotation angles

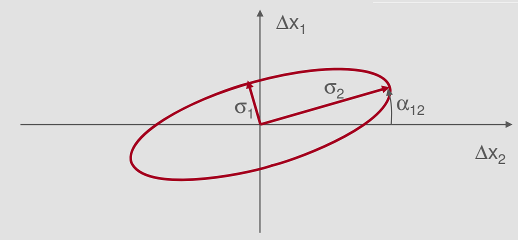
$$x_i' = x_i + \vec{N}(\vec{0}, C')$$ 3.: Mutation of object variables

New convariance matrix C‘ used here!

Operators: Correlated Mutations

- Interpretation of rotation angles α_{ij}
- Mapping onto convariances according to

$$c_{ij(i \neq j)} = \frac{1}{2}(\sigma_i^2 - \sigma_j^2)\tan(2\alpha_{ij})$$

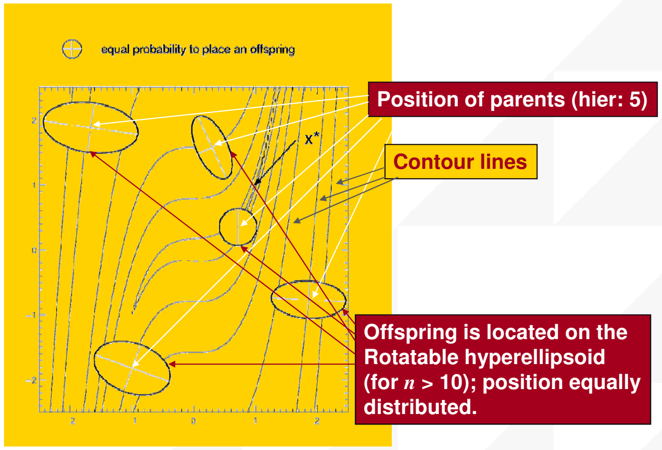


Operators: Correlated Mutation

- τ, τ‘, β are again learning rates
  - τ, τ‘ as before
  - β = 0,0873 (corresponds to 5 degree)
  - Out of boundary correction:

$$|\alpha_j'| > \pi \Rightarrow \alpha_j' \leftarrow \alpha_j' - 2\pi \cdot sign(\alpha_j')$$

Correlated Mutations for ES



Operators: Correlated Mutations

- How to create $\vec{N}(\vec{0},C')$?
  - Multiplication of uncorrelated mutation vector with *n(n-1)/2* rotational matrices

$$\vec{\sigma}_c = \prod_{i=1}^{n-1}\prod_{j=i+1}^{n} R(\alpha_{ij})\cdot\vec{\sigma}_u$$

  - Generates only feasible (positiv definite) correlation matrices

NUTECH SOLUTIONS science for business — Universiteit Leiden

37

Operators: Correlated Mutations

- Structur of rotation matrix

$$R(\alpha_{ij}) = \begin{pmatrix} 1 & & & & & & 0 \\ & 1 & & & & & \\ & & \cos(\alpha_{ij}) & & -\sin(\alpha_{ij}) & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & \sin(\alpha_{ij}) & & \cos(\alpha_{ij}) & & \\ 0 & & & & & 1 & \\ & & & & & & 1 \end{pmatrix}$$

NUTECH SOLUTIONS science for business — Universiteit Leiden

38

Operators: Correlated Mutations

- Implementation of correlated mutations

```
nq := n(n-1)/2;
for i:=1 to n do
  σu[i] := σ[i] * Ni(0,1);
for k:=1 to n-1 do
  n1 := n-k;
  n2 := n;
  for i:=1 to k do
      d1     := σu[n1]; d2:= σu[n2];
      σu[n2] := d1*sin(α[nq])+ d2*cos(α[nq]);
      σu[n1] := d1*cos(α[nq])- d2*sin(α[nq]);
      n2     := n2-1;
      nq     := nq-1;
  od
od
```

Generation of the uncorrelated mutation vector

Rotations

NUTECH SOLUTIONS science for business — Universiteit Leiden

39

Pros and Cons: Correlated Mutations

- Advantages:
  - Individual scaling of object variables
  - Rotation of coordinate system possible
  - Increased global convergence reliability
- Disadvantages:
  - Much slower convergence
  - Effort for mutations scales quadratically
  - Self-adaptation very inefficient

NUTECH SOLUTIONS science for business — Universiteit Leiden

40

Operators: Mutation – Addendum

- Generating N(0,1)-distributed random numbers?

$$u = 2 \cdot U[0,1) - 1$$
$$v = 2 \cdot U[0,1) - 1$$
$$w = u^2 + v^2$$
$$\left.\begin{aligned} x_1 &= u \cdot \sqrt{\frac{-2\log(w)}{w}} \\ x_2 &= v \cdot \sqrt{\frac{-2\log(w)}{w}} \end{aligned}\right\} \text{If } w > 1$$

$$x_1, x_2 \sim N(0,1)$$

NuTech Solutions science for business · Universiteit Leiden · 41

Evolution Strategy:

Algorithms Recombination

NuTech Solutions science for business · Universiteit Leiden · 42

Operators: Recombination

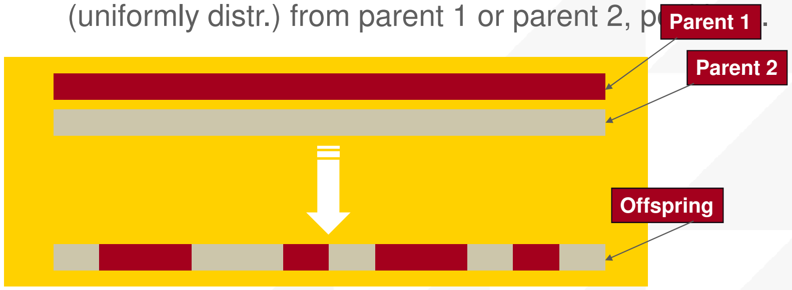
- Only for μ > 1
- Directly after Secektion
- Iteratively generates λ offspring:

```
for i:=1 to λ do
   choose recombinant r1 uniformly at random
        from parent_population;
   choose recombinant r2 <> r1 uniformly at random
        from parent population;
   offspring := recombine(r1,r2);
   add offspring to offspring_population;
od
```

NuTech Solutions science for business · Universiteit Leiden · 43

Operators: Recombination

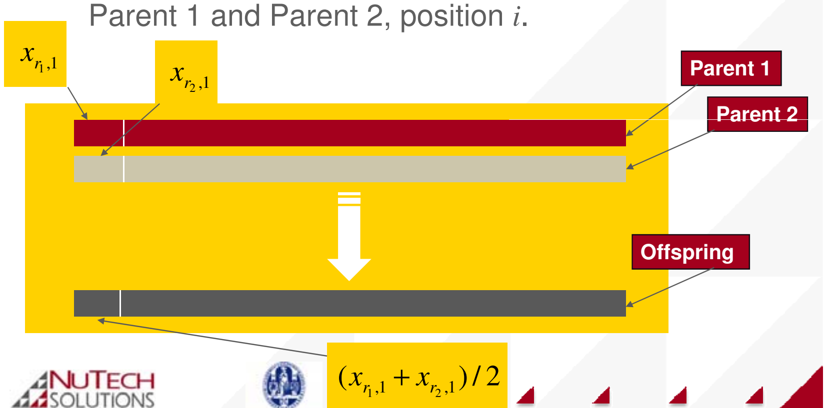
- How does recombination work?
- Discrete recombination:
  - Variable at position i will be copied at random (uniformly distr.) from parent 1 or parent 2, p .



NuTech Solutions science for business · Universiteit Leiden · 44

Operators: Recombination

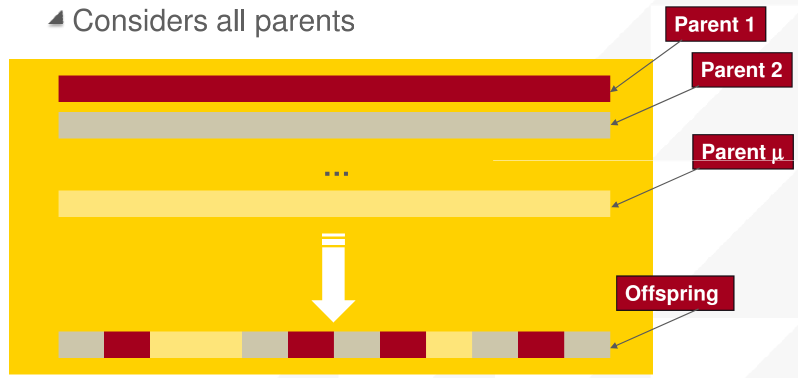
- Intermediate recombination:
  - Variable at position i is arithmetic mean of Parent 1 and Parent 2, position i.



45

Operators: Recombination

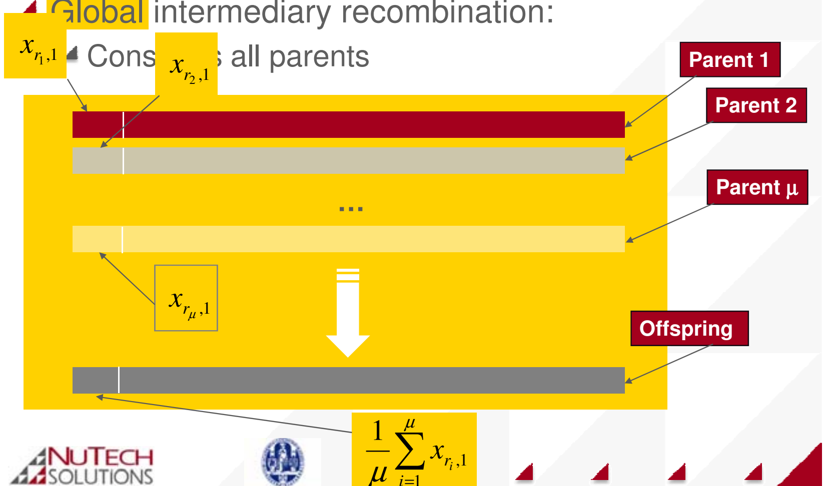
- Global discrete recombination:
  - Considers all parents



46

Operators: Recombination

- Global intermediary recombination:
  - Considers all parents



47

Evolution Strategy

Algorithms Selection

48

Operators: (μ+λ)-Selection

= Actual solution candidate

= New solution candidate

- (μ+λ)-Selection means:
  - μ parents produce λ offspring by
    - (Recombination +)
    - Mutation
  - These μ+λ individuals will be considered together
  - The μ best out of μ+λ will be selected („survive")
    - Deterministic selection
  - This method guarantees monotony
    - Deteriorations will never be accepted

Recombination may be left out
Mutation always exists!

Universiteit Leiden

49

Operators: (μ,λ)-Selection

- (μ,λ)-Selection means:
  - μ parents produce λ >> μ offspring by
    - (Recombination +)
    - Mutation
  - λ offspring will be considered alone
  - The μ best out of λ offspring will be selected
    - Deterministic selection
  - The method doesn't guarantee monotony
    - Deteriorations are possible
    - The best objective function value in generation $t+1$ may be worse than the best in generation t.

Universiteit Leiden

50

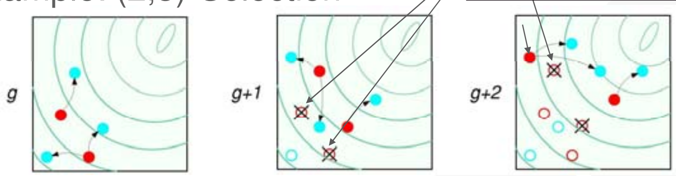
Operators: Selection

Parents don't survive …

Parents don't survive!

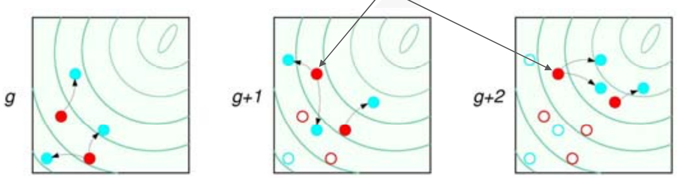
… but a worse offspring.

- Example: (2,3)-Selection



- Example: (2+3)-Selection

… now this offspring survives.



Universiteit Leiden

51

Operators: Selection

Exception!

- Possible occurrences of selection
  - (1+1)-ES: One parent, one offspring, 1/5-Rule
  - (1,λ)-ES: One Parent, λ offspring
    - Example: (1,10)-Strategy
    - One step size / n self-adaptive step sizes
    - Mutative step size control
    - Derandomized strategy
  - (μ,λ)-ES: μ > 1 parents, λ > μ offspring
    - Example: (2,15)-Strategy
    - Includes recombination
    - Can overcome local optima
  - (μ+λ)-strategies: elitist strategies

Universiteit Leiden

52

Evolution Strategy:

Self adaptation of step sizes

Self-adaptation

- No deterministic step size control!
- Rather: Evolution of step sizes
  - Biology: Repair enzymes, mutator-genes
- Why should this work at all?
  - Indirect coupling: step sizes – progress
  - Good step sizes improve individuals
  - Bad ones make them worse
  - This yields an indirect step size selection

Self-adaptation: Example

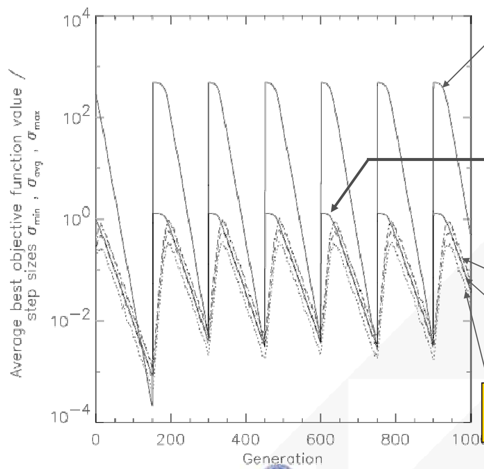
- How can we test this at all?
- Need to know optimal step size …
  - Only for very simple, convex objective functions
  - Here: Sphere model

$\vec{x}^*$: Optimum

$$f(\vec{x}) = \sum_{i=1}^{n} (x_i - x_i^*)^2$$

  - Dynamic sphere model
    - Optimum locations changes occasionally

Self-adaptation: Example



Objective function value

According to theory ff optimal step sizes

$$\sigma_{opt} = c_{\mu,\lambda} \frac{R}{n}$$

Largest …

average …

… and smallest step size measured in the population

Self-adaptation

- Self-adaptation of one step size
  - Perfect adaptation
  - Learning time for back adaptation proportional n
  - Proofs only for convex functions
- Individual step sizes
  - Experiments by Schwefel
- Correlated mutations
  - Adaptation much slower

57

Evolution Strategy:

Derandomization

58

Derandomization

- Goals:
  - Fast convergence speed
  - Fast step size adaptation
  - Precise step size adaptation
  - Compromise convergence velocity – convergence reliability
- Idea: Realizations of $N(0, \sigma)$ are important!
  - Step sizes and realizations can be much different from each other
  - Accumulates information over time

59

Derandomzed (1,λ)-ES

- Current parent: $\vec{x}^g$ in generation g
- Mutation ($k=1,\dots,\lambda$):

Offspring k

Global step size in generation g

$$\vec{x}^g_{N_k} = \vec{x}^g + \delta^g \cdot \vec{\delta}^g_{scal} \cdot \vec{Z}_k$$

$$\vec{Z} = (z_1,\dots,z_n) \qquad z_i \sim N(0,1)$$

Individual step sizes in generation g

- Selection: Choice of best offspring

$$\vec{x}^{g+1} = \vec{x}^g_{N_{sel}}$$

Best of λ offspring in generation g

60

Derandomized (1,λ)-ES

- Accumulation of selected mutations:

$$\vec{Z}_A^{g} = (1-c)\cdot \vec{Z}_A^{g-1} + c\cdot \vec{Z}_{sel}$$

The particular mutation vector, which created the parent!

- Also: weighted history of good mutation vectors!
- Initialization:

$$\vec{Z}_A^{0} = \vec{0}$$

- Weight factor:

$$c = \frac{1}{\sqrt{n}}$$

Derandomized (1,λ)-ES

- Step size adaptation:

Norm of vector

$$\delta^{g+1} = \delta^{g}\cdot\left(\exp\left(\frac{\left|\vec{Z}_A^{g}\right|}{\sqrt{n}\cdot\sqrt{\frac{c}{2-c}}}\right)-1+\frac{1}{5n}\right)^{\beta}$$

Vektor of absolute value

$$\vec{\delta}_{scal}^{g+1} = \vec{\delta}_{scal}^{g}\cdot\left(\frac{\left|\vec{Z}_A^{g}\right|}{\sqrt{\frac{c}{2-c}}}+0.35\right)^{\beta_{scal}}$$

Regulates adaptation speed and precision

Derandomized (1,λ)-ES

- Explanations:
  - Normalization of average variations in case of missing selection (no bias):

$$\sqrt{\frac{c}{2-c}}$$

  - Correction for small n: $1/(5n)$
  - Learning rates:

$$\beta = \sqrt{1/n}$$
$$\beta_{scal} = 1/n$$

Evolution Strategy:

Rules of thumb

Some Theory Highlights

- Convergence velocity:

  Problem dimensionality

  $\varphi \sim 1/n$

- For (1,λ)-strategies:

  $\varphi \sim \ln \lambda$

  Speedup by λ is just logarithmic – more processors are only to a limited extend useful to increase φ.

- For (μ,λ)-strategis (discrete and intermediary recombination): $\varphi \sim \mu \ln \frac{\lambda}{\mu}$

  Genetic Repair Effect of recombination!

…

- For strategies with global intermediary recombination:

  $\lambda = 4 + \lfloor 3 \log n \rfloor$
  $\mu = \lfloor \lambda / 2 \rfloor$

- Good heuristic for (1,λ):

  $\lambda = 10$

n	λ	μ
10	10.91	5.45
20	12.99	6.49
30	14.20	7.10
40	15.07	7.53
50	15.74	7.87
60	16.28	8.14
70	16.75	8.37
80	17.15	8.57
90	17.50	8.75
100	17.82	8.91
110	18.10	9.05
120	18.36	9.18
130	18.60	9.30
140	18.82	9.41
150	19.03	9.52

Mixed-Integer Evolution Strategies

Mixed-Integer Evolution Strategy

- Generalized optimization problem:

$$f(r_1, \ldots, r_{n_r}, z_1, \ldots, z_{n_z}, d_1, \ldots, d_{n_d}) \rightarrow min$$

subject to:

$$r_i \in [r_i^{min}, r_i^{max}] \subset \mathbb{R}, \;\; i = 1, \ldots, n_r$$
$$z_i \in [z_i^{min}, z_i^{max}] \subset \mathbb{Z}, \;\; i = 1, \ldots, n_z$$
$$d_i \in D_i = \{d_{i,1}, \ldots, d_{i,|D_i|}\}, i = 1, \ldots, n_d$$

Mixed-Integer ES: Mutation

for $i = 1, \ldots, n_r$ do
  $s_i' \leftarrow s_i \exp(\tau_g N_g + \tau_l N(0,1))$
  $r_i' = r_i + N(0, s_i')$
end for
for $i = 1, \ldots, n_z$ do
  $q_i' \leftarrow q_i \exp(\tau_g N_g + \tau_l N(0,1))$
  $z_i' \leftarrow z_i + G(0, q_i')$
end for
$p_i' := 1/[1 + \frac{1-p_i}{p_i} * \exp(-\tau_l * N(0,1))]$
for $i \in \{1, \ldots, n_d\}$ do
  if $U(0,1) < p_i'$ then
    $d_i' \leftarrow$ uniformly randomly value from D_i
  end if
end for

Learning rates (global)

Learning rates (global)

Geometrical distribution

Mutation Probabilities

Literature

- H.-P. Schwefel: *Evolution and Optimum Seeking*, Wiley, NY, 1995.
- I. Rechenberg: *Evolutionsstrategie 94*, frommann-holzboog, Stuttgart, 1994.
- Th. Bäck: *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, NY, 1996.
- Th. Bäck, D.B. Fogel, Z. Michalewicz (Hrsg.): *Handbook of Evolutionary Computation, Vols. 1,2*, Institute of Physics Publishing, 2000.