Evolutionary Computation

Prof. dr. Thomas Bäck
Natural Computing Group

May 7th, 2007

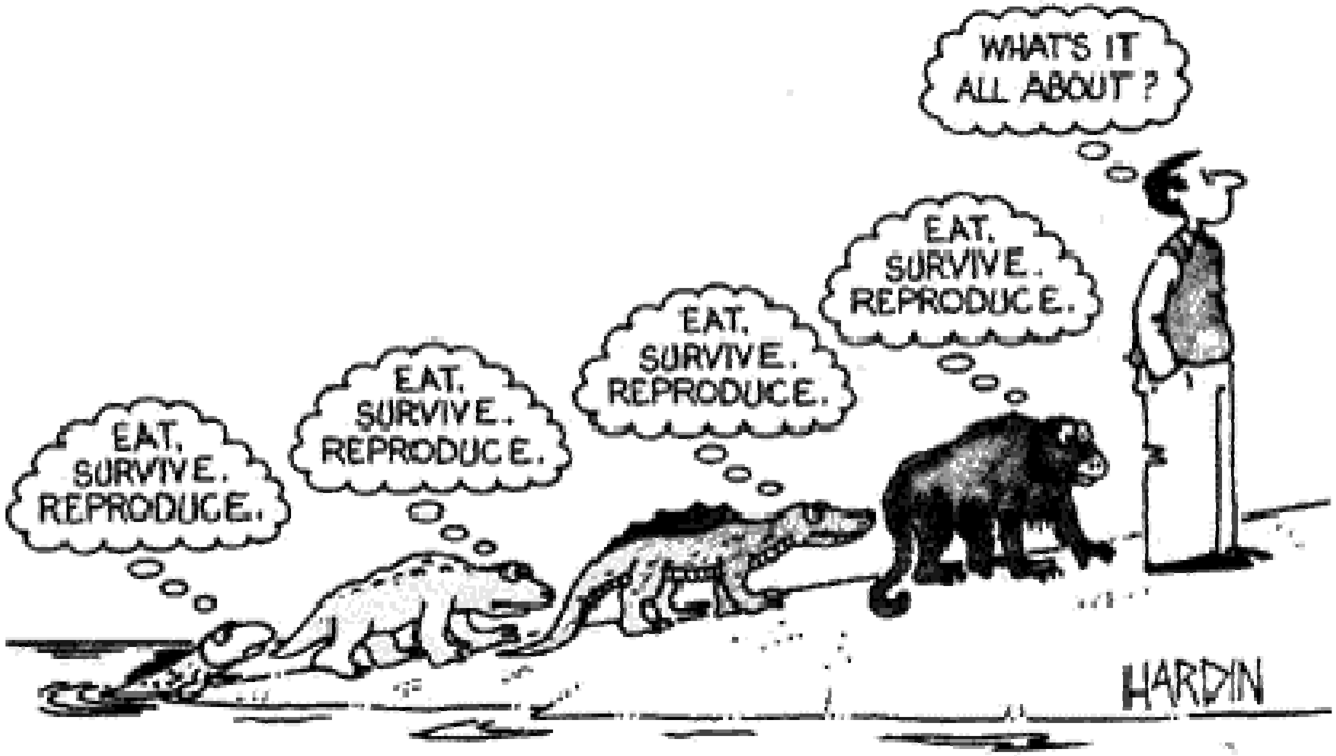
Overview

- Introduction: Evolutionary Algorithms
- Real-World Applications
- Basic Genetic Algorithms
- An Application Problem
  - Low autocorrelation sequences
  - An example algorithm
- A small Research Task – for You!

Introduction

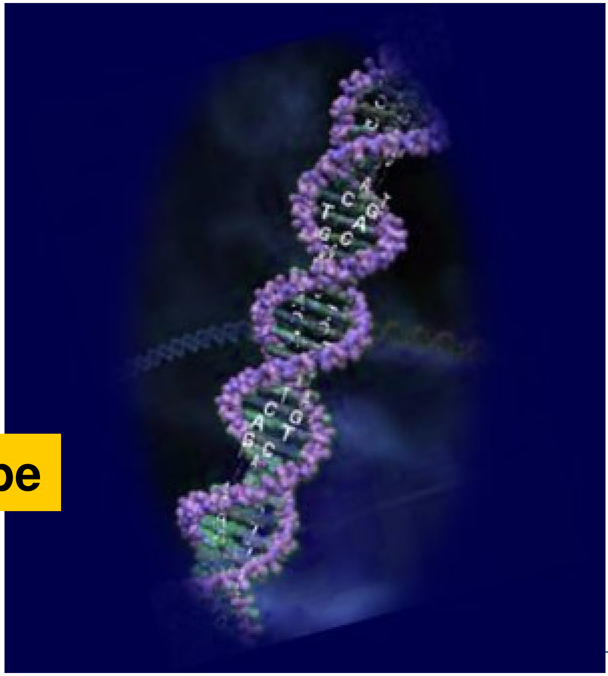
Background I

Biology = Engineering (Daniel Dennett)



Background II

- DNA-Molecule
- Substrate for genetic information
- Human being
  - $3 \cdot 10^9$ base pairs
  - $4^{3 \cdot 10^9}$ combinations



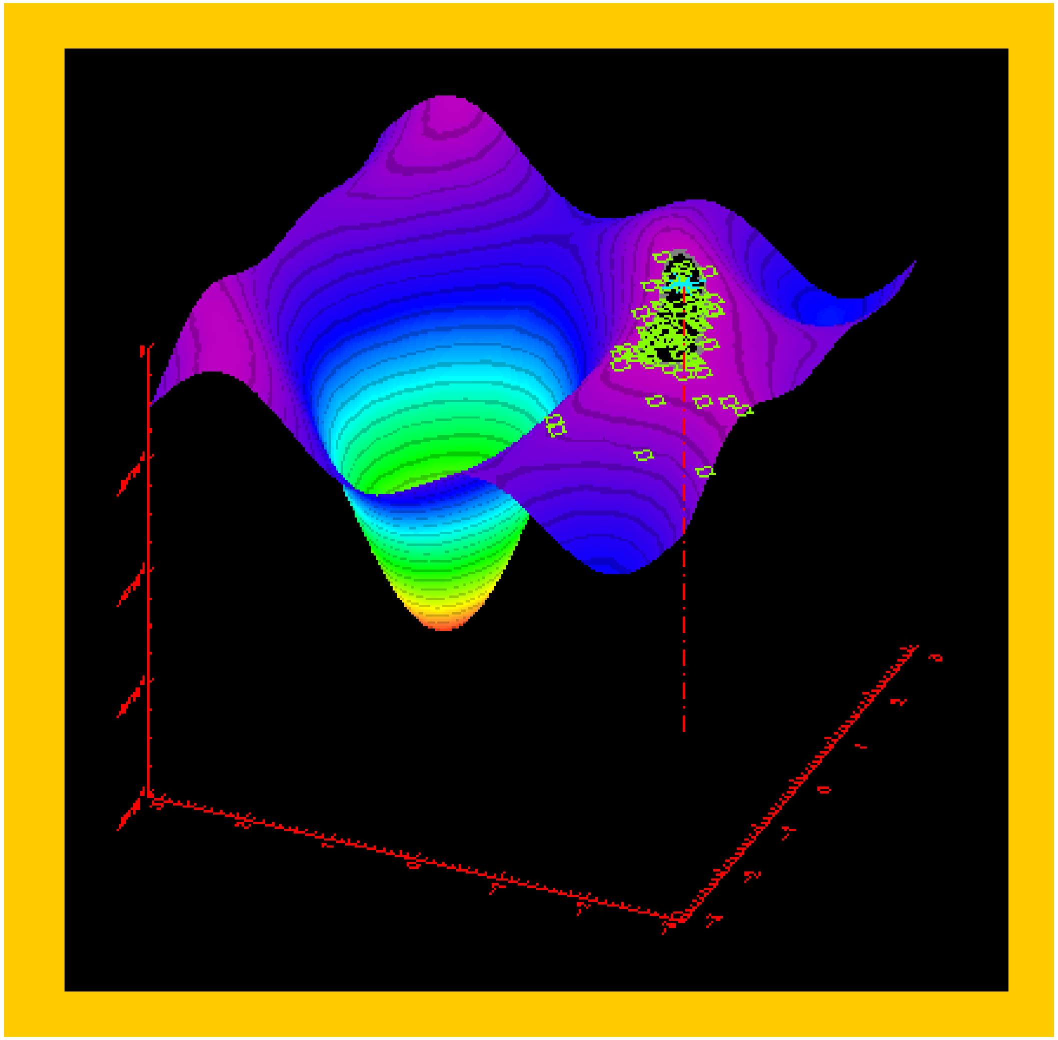
Genotype

Thought Experiment

Globular Cluster NGC 6397
(ESO/MPI 2.2-m + WFI)
ESO PR Photo 23a/04 (17 August 2004)
© European Southern Observatory

- 10^{40} Time steps since Big Bang
- 10^{80} Elementary particles in the universe
- 10^{120} „Computational steps“ in the universe …
- $4^{3 \cdot 10^9}$ is considerable larger!

Optimum tracking of an ES

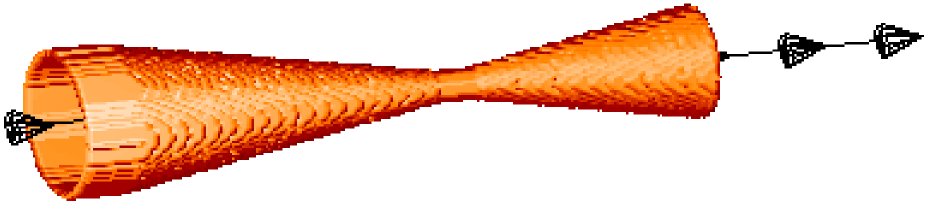


- Dynamic function
- 30-dimensional
- 3D-projection

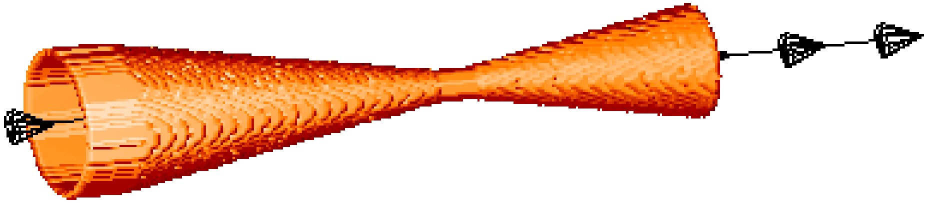
Real-World Applications

Creating Innovation

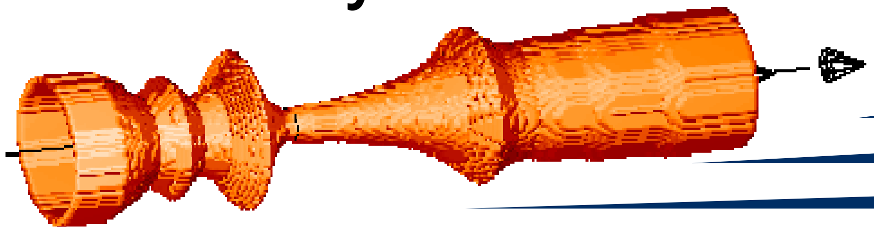
- Illustrative Example: Optimize Efficiency
  - Initial:



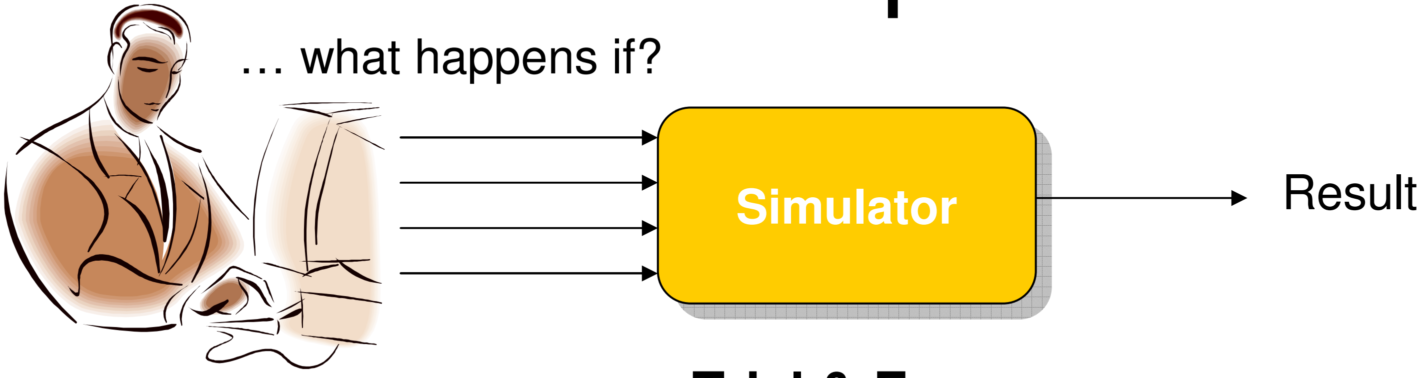
  - Evolution:



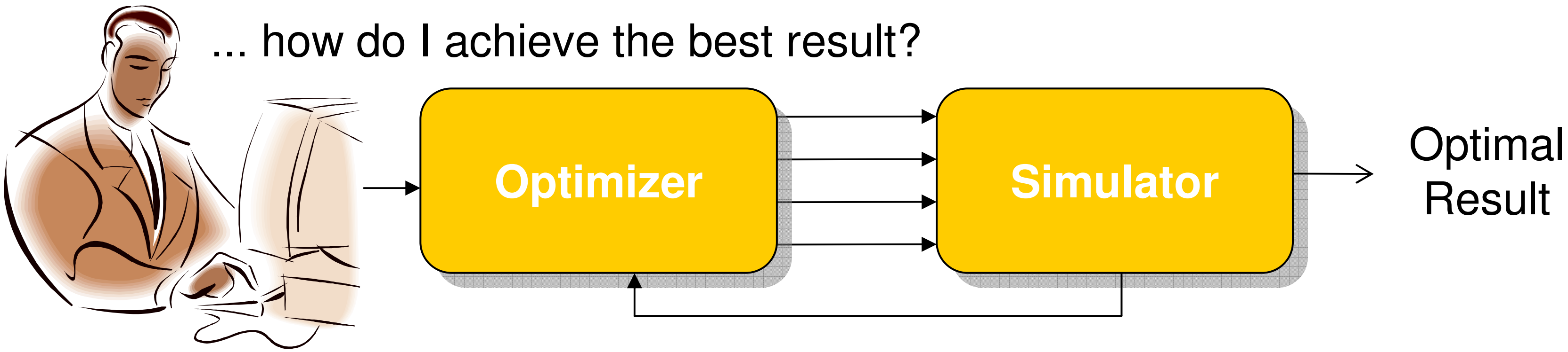
- 32% Improvement in Efficiency !



Simulation vs. Optimization

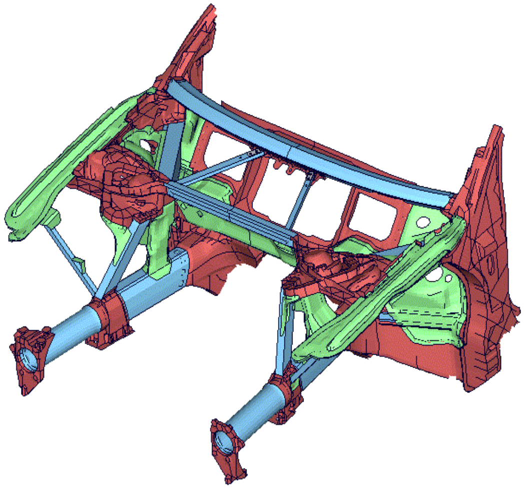


Trial & Error



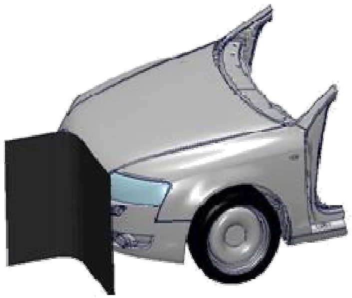
Maximization / Minimization
If so, multiple objectives

Multi-Disziplinary Optimization ASF®- Front part of R8 body



- Pre-optimized Space-Frame-Concept
- Does a better design exist?
- **Aim: Minimization of structural weight**
- **Degrees of freedom**
  - Wall thicknesses of the semi-finished products sheet & profile
  - Material characteristic profile

Considered Disciplines



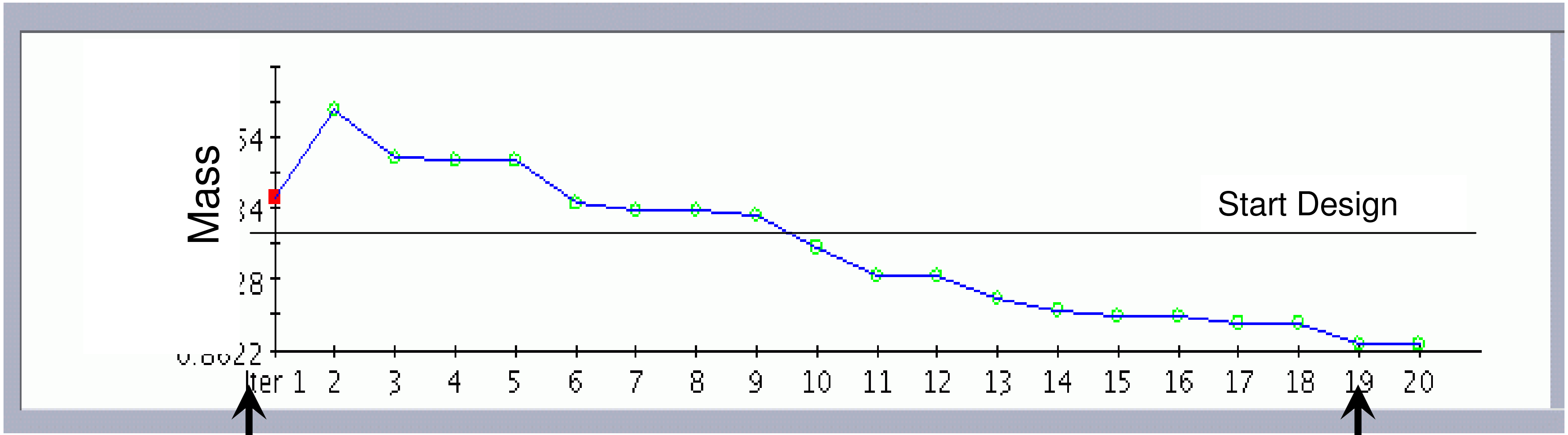
Damage according
to insurance classification,
Component Model,
2 CPUs

Global dynamic stiffness,
Trimmed Body,
1 CPU

Front Crash (EURO NCAP),
Complete Body
4 CPUs

Resources per Design: 7 CPUs, approx. 23h

Course of Optimization



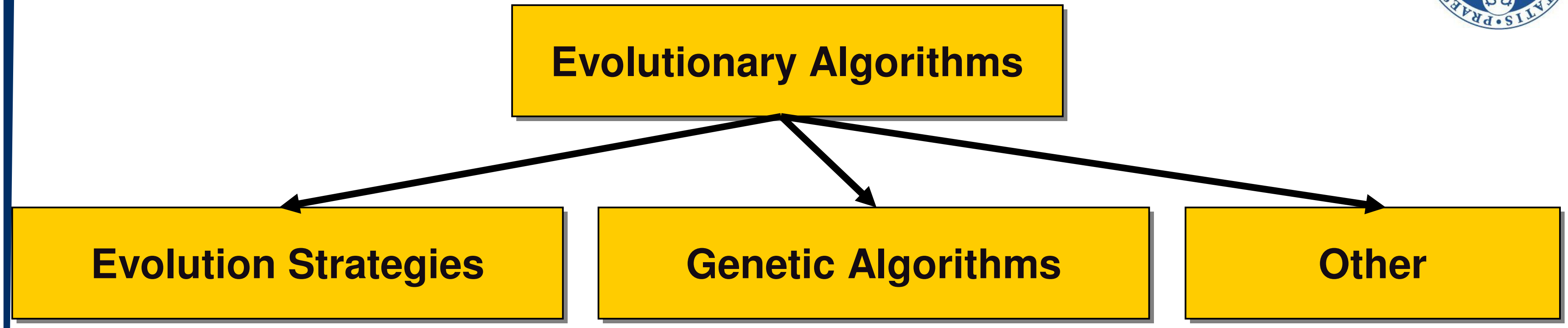
Start design (basis)
1 restriction violated

Weight Reduction

ca. 1 kg

Optimum (Exp. 376)
Restrictions satisfied

Taxonomy of EAs

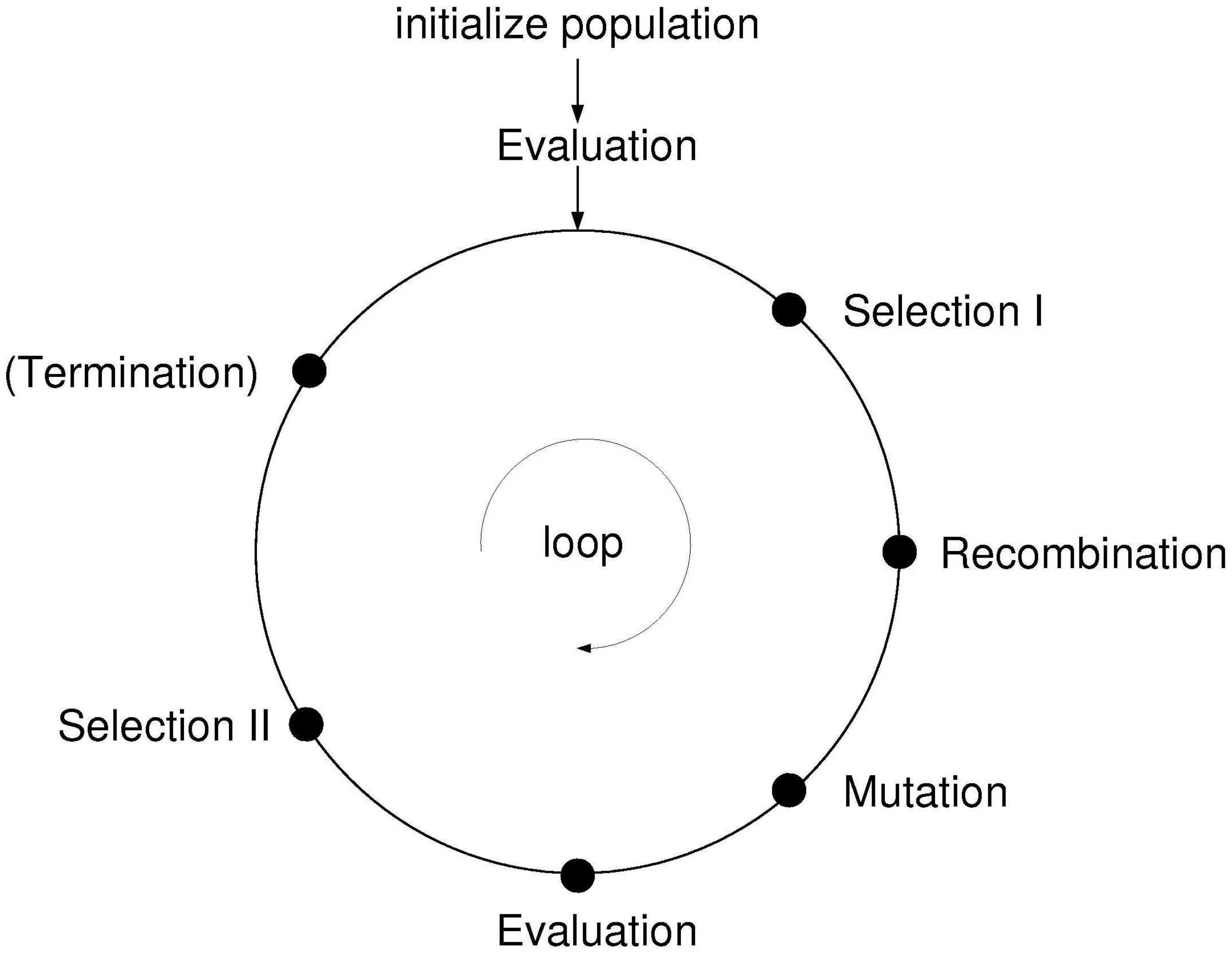


- Mixed-integer capabilities
- Emphasis on mutation
- Self-adaptation
- Small population sizes
- Deterministic selection
- Developed in Germany
- Theory focused on convergence velocity

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

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

Evolutionary Loop



Basic Genetic Algorithms

Basic Genetic Algorithms (GA)

- Using binary representation $\{0,1\}^n$
- Variation operators:
  - Mutation (small change of one individual)
  - Crossover (exchange between two individuals)
- Selection (probabilistic by fitness)
- Population size constant, big

Basic GA: Mutation

- Mutation rate $p_m \in [0,1]$

1	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1

$$\downarrow$$

1	0	0	0	1	1	1	1	1	1	1	0	1	0	1	1

- Mutation rate is often very small (e.g., $1/n$)

Basic GA: Crossover

- Crossover rate $p_c \in [0,1]$

1	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1
0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1

1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	1
0	0	0	1	1	0	0	1	0	0	0	0	1	0	1	1

- Randomly chosen crossover point

Basic GA: Selection

- Older method: Proportional selection

$$p(\vec{x}_i) = \frac{f(\vec{x}_i)}{\sum_{j=1}^{N} f(\vec{x}_j)}$$

- N population size
- Gives probability of individual i to be selected

Basic GA: Selection

- Disadvantages!
- Works only for maximization
- Sometimes too small probability differences
- Implementation nontrivial

Tournament Selection

For $i=1,...N$:

- Select q individuals at random from population
- Copy best of those q into the new generation

- Typical tournament size, e.g., $q = 2$
- $q = 1$ does not work (why ?)

Other Selection Operators

- As an alternative: Varying population size
- μ parents, λ offspring, $\lambda >> \mu$ (15, 100)
- (μ,λ)-Selection:
  - μ parents generate λ offspring (mutation, crossover)
  - Out of λ offspring, the best μ are selected to survive
  - Example: (15, 100)-selection, (1,10)-selection

Other Selection Operators

- $(\mu+\lambda)$-Selection:
  - μ parents generate λ offspring (mutation, crossover)
  - Out of λ offspring **and the μ parents**, the best μ are selected to survive
  - Example: (15+100)-selection, (1+10)-selection
- What does it mean for evolution?

An Application Problem: LABS

Example Problem: LABS

- Low autocorrelation binary sequences
- Autocorrelation function on $\{-1,1\}^n$
- Important applications
  - Telecommunications
  - Radar
  - Sonar
- Transformation of variables:
  - $\{0,1\} \rightarrow \{-1,1\}$

$$y_i = 2x_i - 1$$

The Objective Function

- Search space: $\{0,1\}^n$
- Goal: Find $\vec{x} \in \{0,1\}^n$ such that

$$E(\vec{x}) = \left(\sum_{k=1}^{n-1} \left(\sum_{i=1}^{n-k} y_i \cdot y_{i+k} \right)^2 \right) \rightarrow \min$$

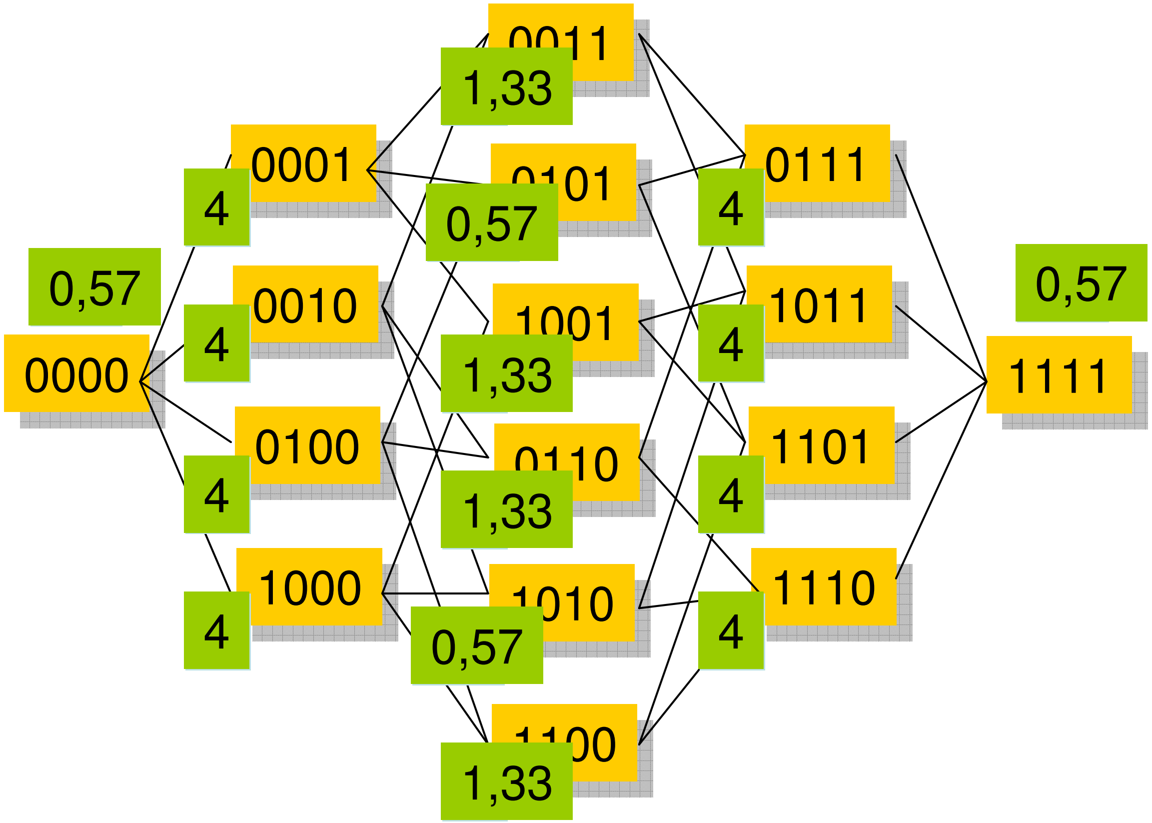
The Objective Function

- Alternative Formulation:

$$F(\vec{x}) = \frac{n^2}{2E(\vec{x})} \rightarrow \max$$

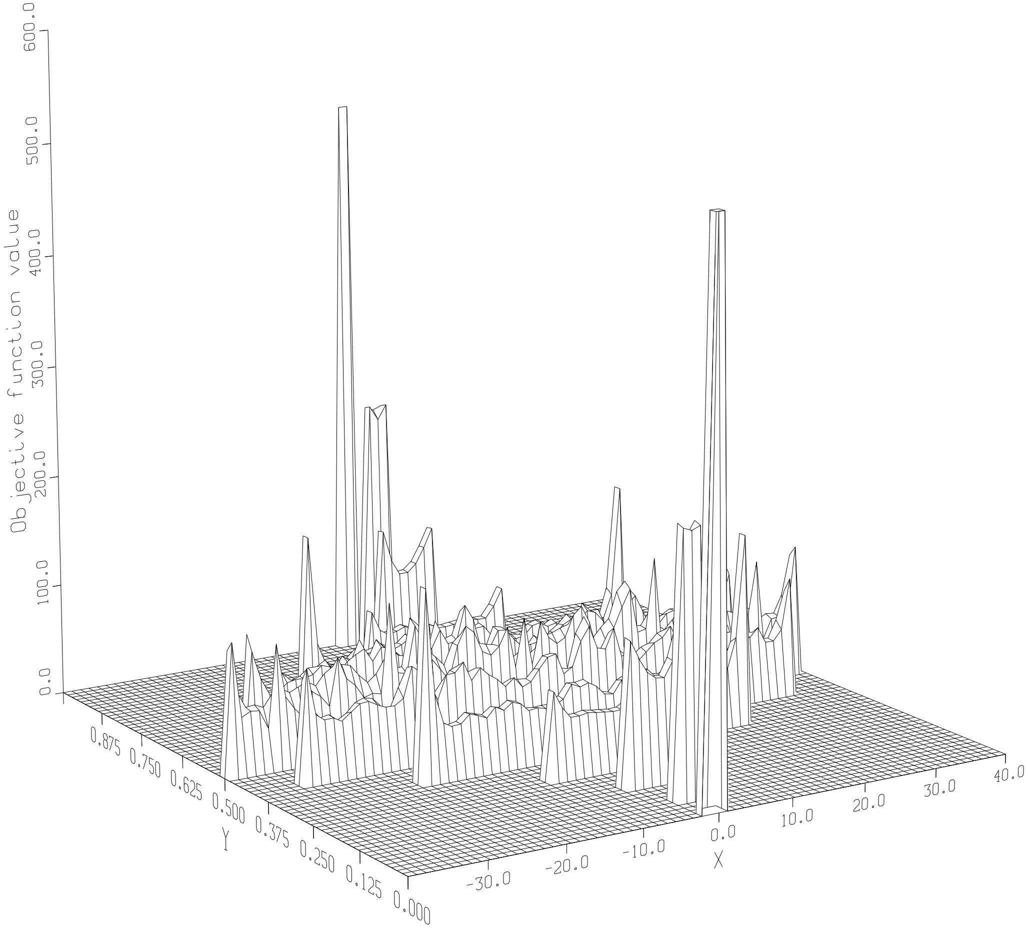
- Merit Factor

Example: $n=4$



$$E(\vec{x}) = (y_1 y_2 + y_2 y_3 + y_3 y_4)^2 + (y_1 y_3 + y_2 y_4)^2 + (y_1 y_4)^2$$

Example: $n=12$



Objective function value
0.0
100.0
200.0
300.0
400.0
500.0
600.0
Y
0.875
0.750
0.625
0.500
0.375
0.250
0.125
0.000
X
-30.0
-20.0
-10.0
0.0
10.0
20.0
30.0
40.0

Some Values ...

- Theory indicates that

$$\lim_{n\to\infty} \arg\max F(\vec{x}) \approx 12{,}32$$

- See table for some known records
  - Values in bold are not confirmed to be best possible
  - Most optimizers get stuck around 7

n	Best value of f
20	7.6923
50	8.1699
100	**8.6505**
199	**7.5835**
200	**7.4738**
201	**7.5263**
202	**7.3787**
203	**7.5613**
219	**7.2122**
220	**7.0145**
221	**7.2207**
222	**7.0426**

Some Values

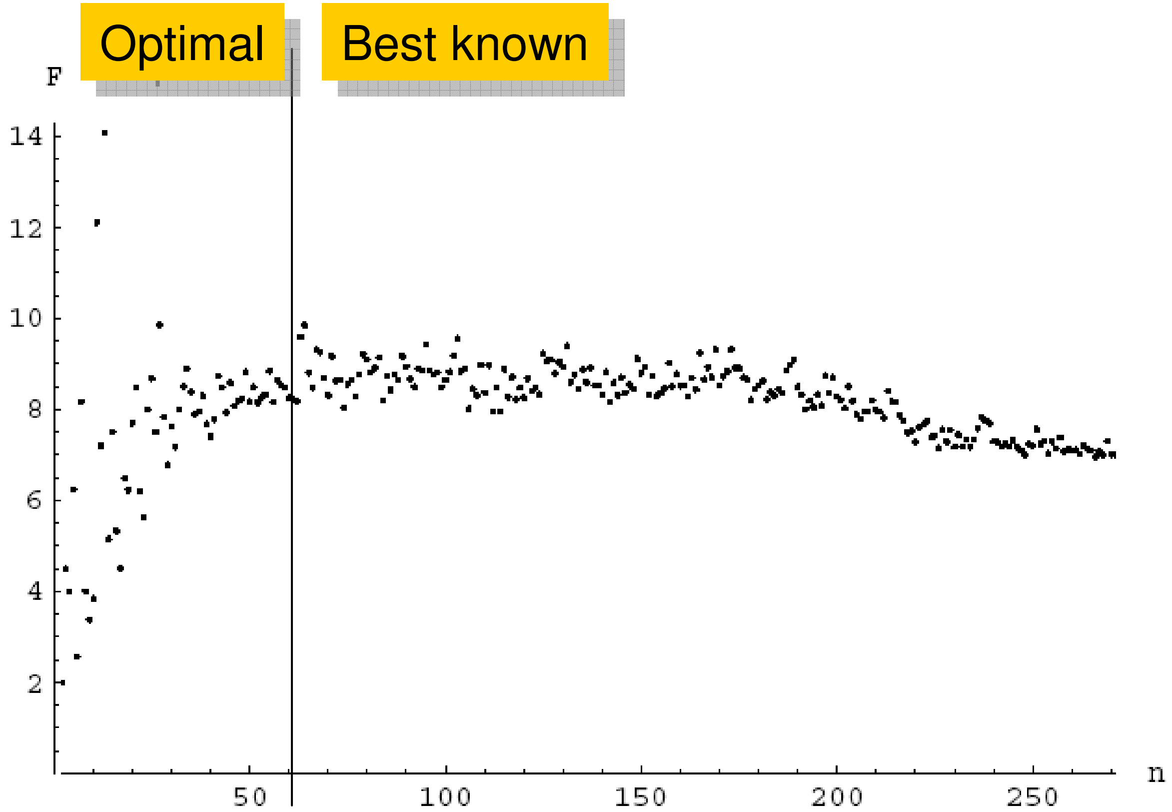


Figure 1: The optimal merit factor (for $2 \leq n \leq 60$) and the best known merit factor (for $61 \leq n \leq 271$) for binary sequences of length n.

An Algorithm

De Groot et al, 1989

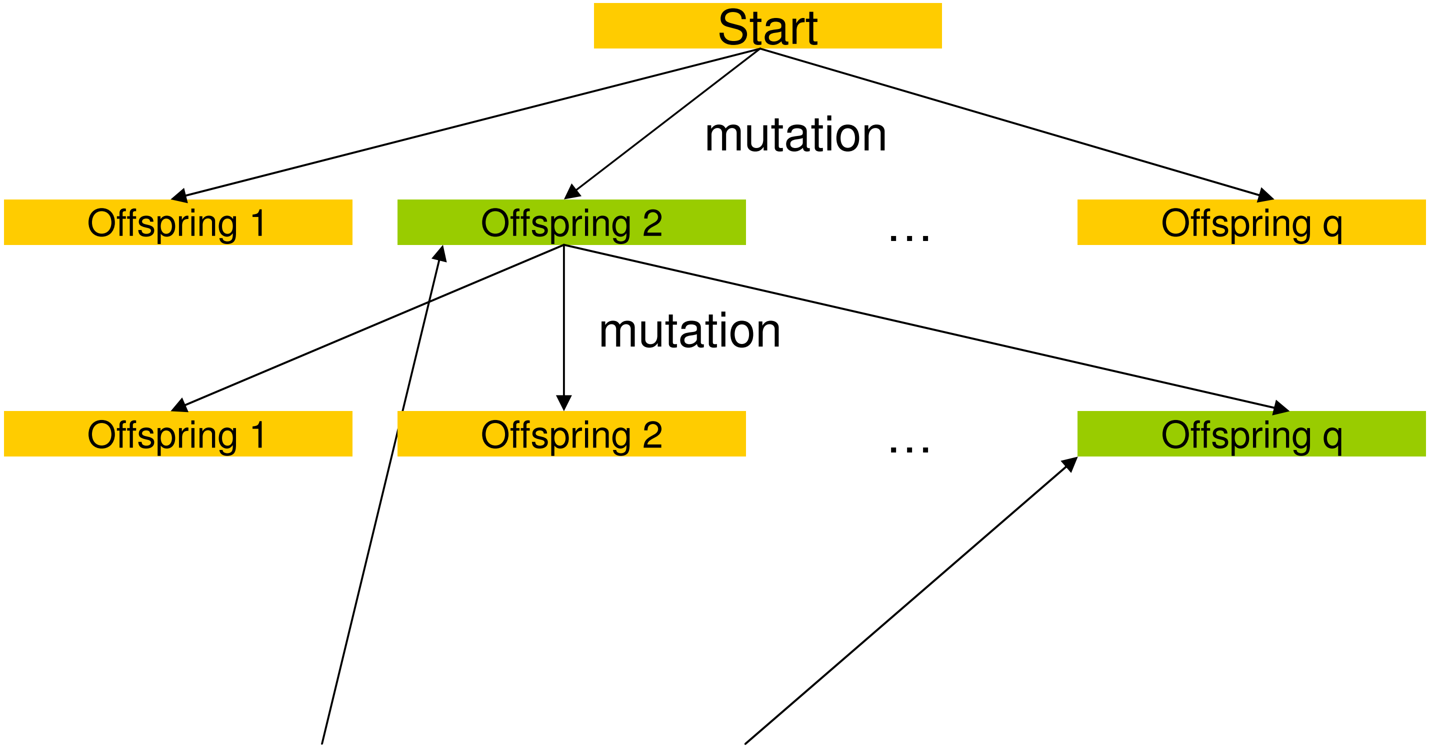
1. N individuals (randomly); evaluate
2. Per individual $\vec{x}$:
   - Create offspring by mutation($\vec{x}$)
3. Evaluate all offspring
4. Select the best N out of ($N+kN$) parents and offspring
5. If termination criterion not satisfied:
   Go to 2.

Mutation

1. Repeat for $i=1,\ldots k$:
   - Create q individuals by q random 1-bit-changes($\vec{x}_i$)
   - Memorize the best i-bit mutant $\vec{x}_i$
2. Return $\vec{x}_1,\ldots,\vec{x}_k$

Best 1-bit mutant, …, best k-bit mutant

Illustration:



Best 1-bit mutant, best 2-bit mutant, …

Some Results they Found

n	Best-so-far	This algorithm
81	8,20	8,04
101	8,36	8,36
141	6,01	6,48
181	5,70	5,81

- Parameter values: q, k, N
- Not given in their paper

Research Homework

- Implement an evolutionary search algorithm
- Describe the algorithm (general idea)
- Apply the algorithm to LABS
- Report your results for the following n values:
  - 100, 200, 201, 202, 203, 219, 220, 221, 222

Thank You !