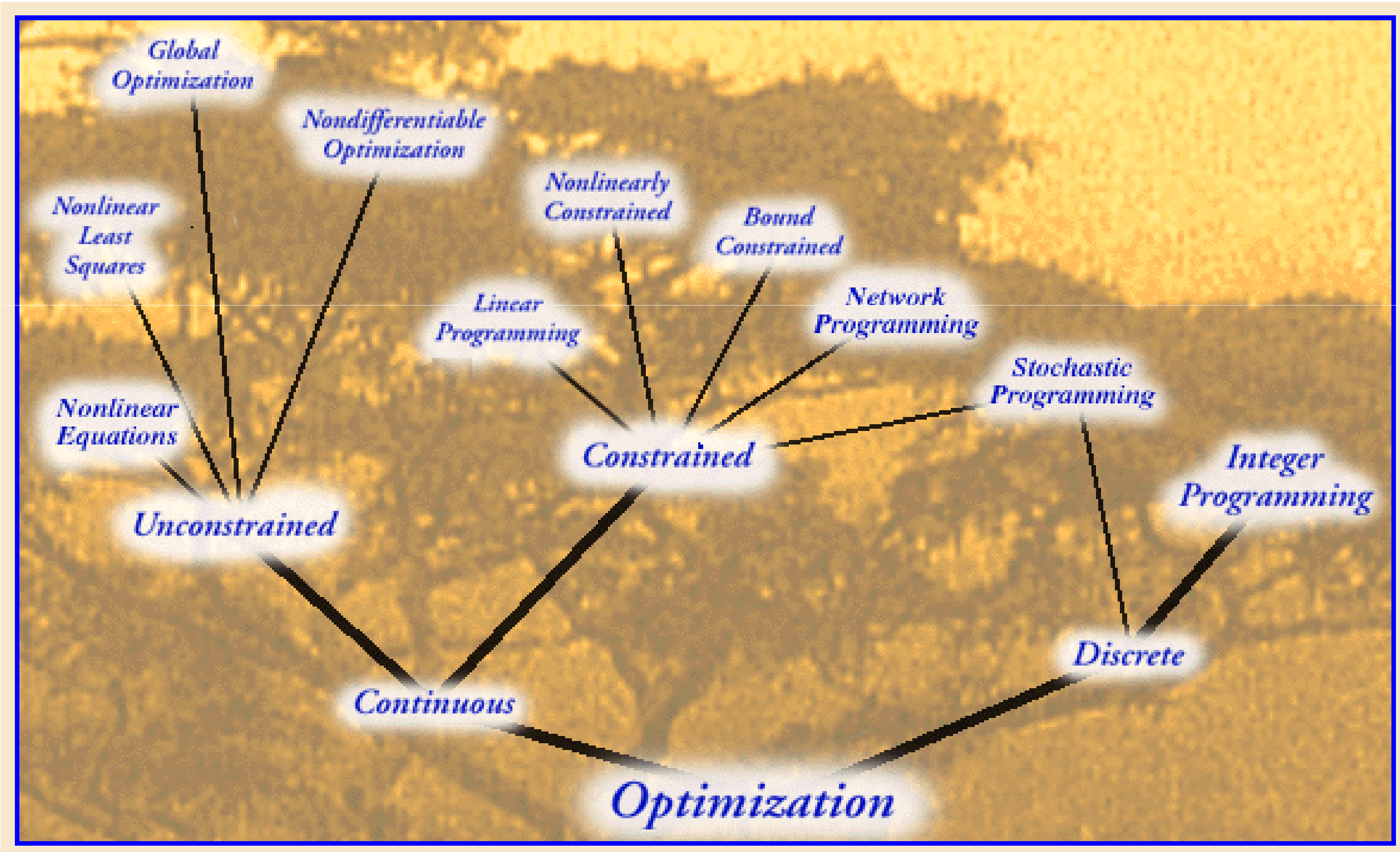
OPTIMIZATION METHODS

For more information visit:
www.esteco.com or send an e-mail to:
modeFRONTIER@esteco.com

NEOS Optimization Tree

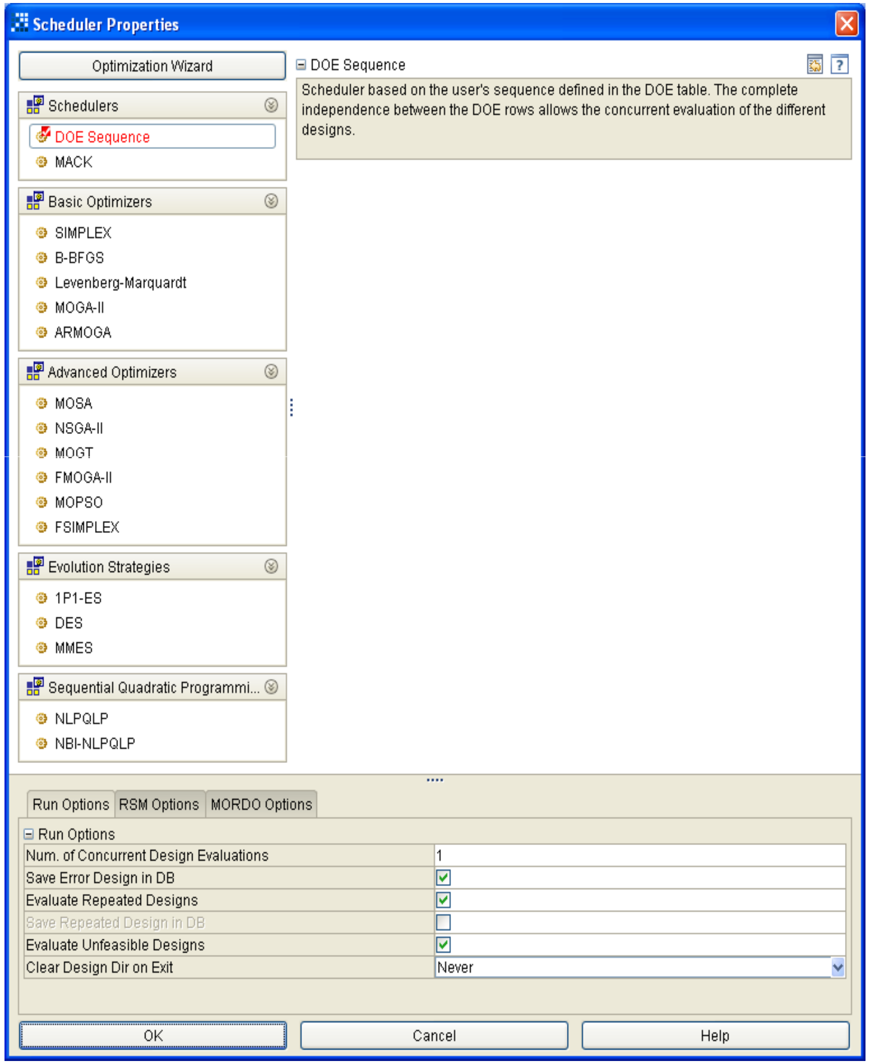


What does the modeFRONTIER support?

Which of the following are supported by modeFRONTIER?

Continuous Nonlinear Program
Continuous Linear Program
Integer Linear Program
Integer Nonlinear Program
Mixed Integer Linear Program
Mixed Integer Nonlinear Program
Multiple Objective Continuous Nonlinear Program
Multiple Objective Mixed Integer Nonlinear Program

modeFRONTIER Optimization Algorithms



Available Algorithms:

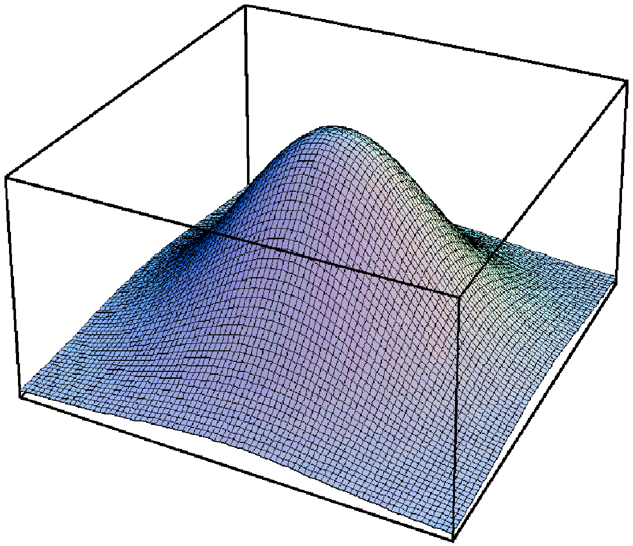
- **Schedulers**
  - DOE Sequence
  - MACK — Multivariate Adaptive Crossvalidating Kriging
- **Basic Optimizers**
  - SIMPLEX — Single-objective derivative-free optimizer
  - B-BFGS — Single objective Bounded BFGS algorithm
  - Levenberg-Marquardt
  - MOGAII — Multi Objective Genetic Algorithm
  - ARMOGA — Adaptive Range MOGA
- **Advanced Schedulers**
  - MOSA — Multi Objective Simulated Annealing Algorithm
  - NSGA II — Non-dominated Sorting Genetic algorithm
  - MOGT — Game Theory coupled with Simplex algorithm
  - F-MOGAII — Fast Multi Objective Genetic Algorithm
  - MOPSO — Multi Objective Particle Swarm Optimizer
  - F-SIMPLEX — Fast Single-objective derivative-free optimizer
- **Evolution Strategy Schedulers**
  - 1P1-ES
  - DES — Derandomised Evolution Strategy
  - MMES — Multi-membered evolution strategy
- **Sequential Quadratic programming**
  - NLPQLP — Robust implementation of a sequential quadratic programming algorithm.
  - NLPQLP-NBI — Multi-objective scheduler based on the NBI - Normal-Boundary Intersection method

Comparison Between Methods

- **Convergence rate :** higher for gradient-based methods (SQP, BFGS), SIMPLEX and MOGT
- **Accuracy :** better for gradient-based methods
- **Robustness :** much higher for probabilistic methods (GA, ES), good for Simplex
- **Multi-objective:** MOGA-II, NSGA-II, F-MOGA-II, MOSA, MOGT, MOPSO, NBI-NLPQLP and MMES

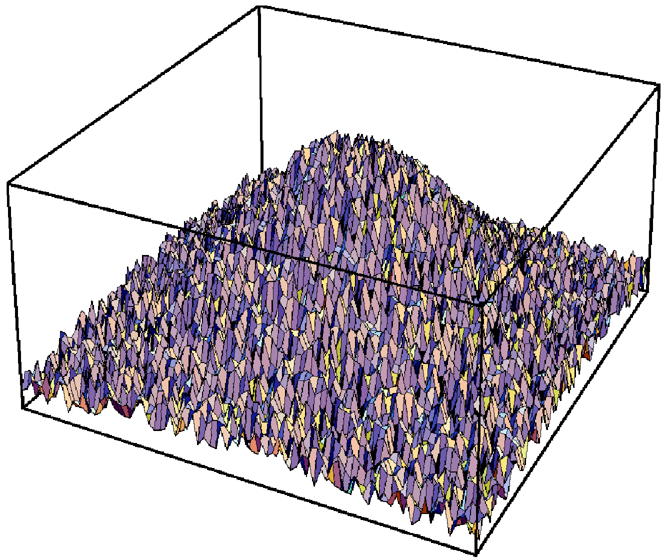
Difficulties of Numerical Optimization

There is a huge difference between mathematical optimization and optimization in the real-world applications



Ideal function in the mathematical world

Rugged hill in the experimental world



Difficulties of Numerical Optimization

1. Most optimization algorithms have difficulty dealing with discontinuous functions.
2. The functions may be unknown (black-boxes) or very complex
3. Computational time increases as the number of design variables increases.
4. Optimization techniques have no stored experience or intuition on which to draw.
5. Most algorithms will get stuck at local optimal points.
6. If the analysis program is not theoretically precise, the results may be misleading.
7. Convergence to optimal solution depends on chosen initial solution.
8. Algorithm efficient in solving one problem may not be efficient in solving a different optimization problem (problem-dependent)
9. Many analysis/simulation programs were not written with automated design in mind.
10. Time consuming functions
11. …

Optimization methods descriptions

- NBI
- SIMPLEX
- Evolution Strategies

5.2 Normal Boundary Intersection (NBI-NLPQLP)

NBI-NLPQLP = Normal Boundary Intersection + NLPQLP

- Multi-objective scheduler based on the NBI method, developed by I. Das and J. E. Dennis.
- The NBI method applies to any smooth multi-objective problem, reducing it to many single-objective constrained subproblems (the "*NBI subproblems*").
- So the NBI method has to be coupled with a single-objective solver in order to solve these subproblems: NLPQLP is used.

NBI-NLPQLP

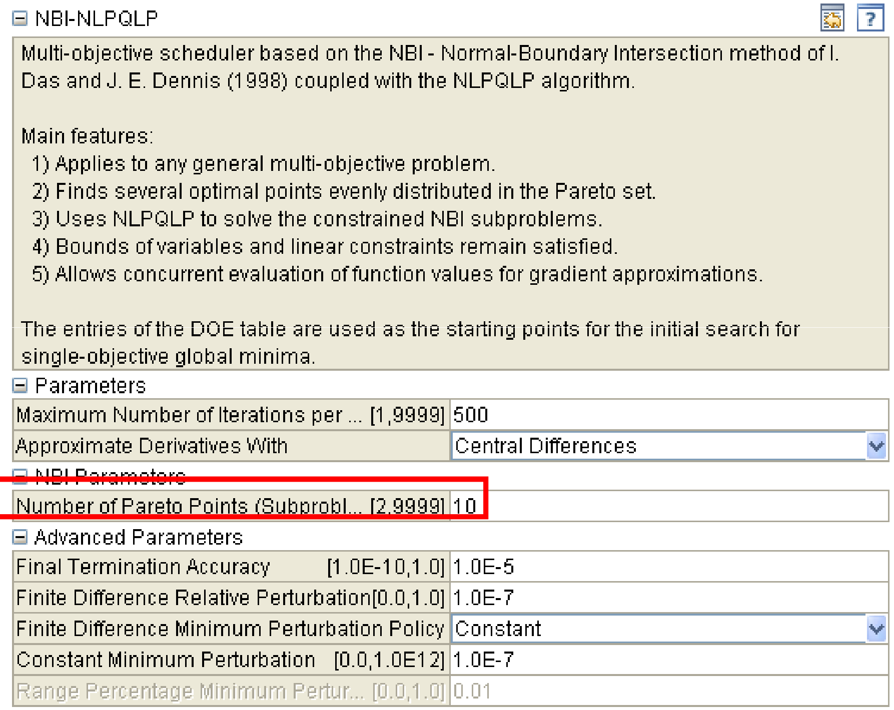
- The **NBI subproblems** are characterized by the introduction of one new variable and *N* constraints, with respect to the original multi-objective problem (*N* = number of objectives).

Algorithm scheme:

1. Evaluation of DOE designs.
2. Each objective function is solved separately, as a single-objective problem (starting from most favorable DOE) → setting of internal parameters.
3. All the NBI subproblems are solved successively.

NLPQLP-NBI Panel

Number of Pareto Points (Sub-problems): larger values imply a better resolution of the Pareto frontier (but request more and more design evaluations).



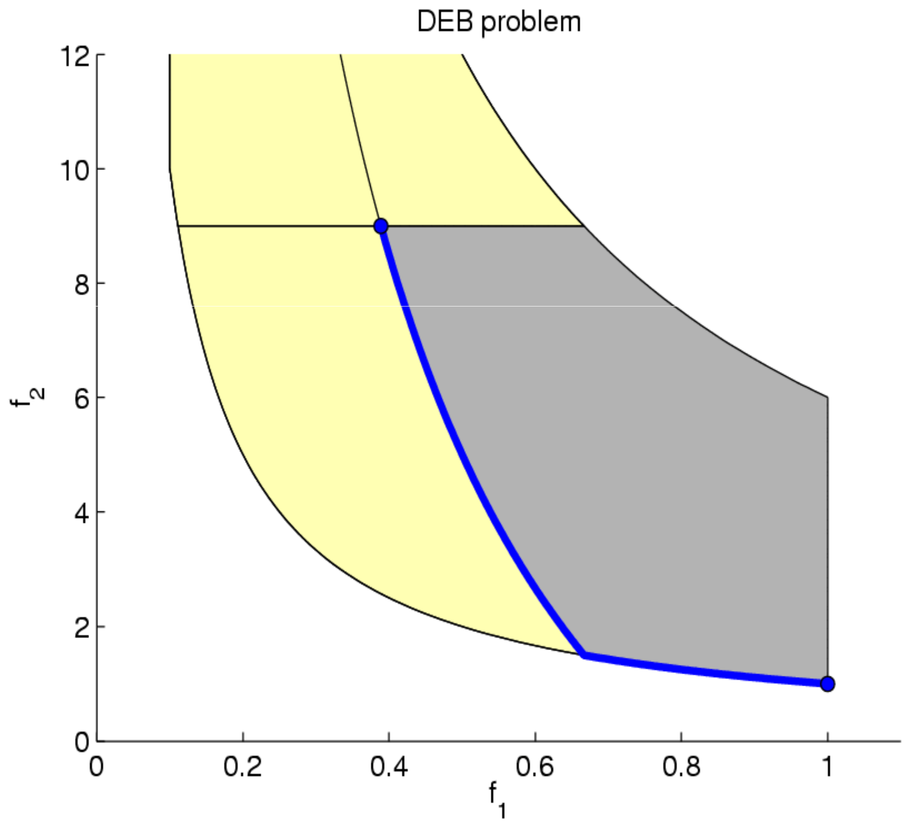
NLPQLP-NBI - Example

Example: DEB problem

$$
\begin{aligned}
\min f_1 &= x_1 \\
\min f_2 &= (1 + x_2)/x_1
\end{aligned}
$$

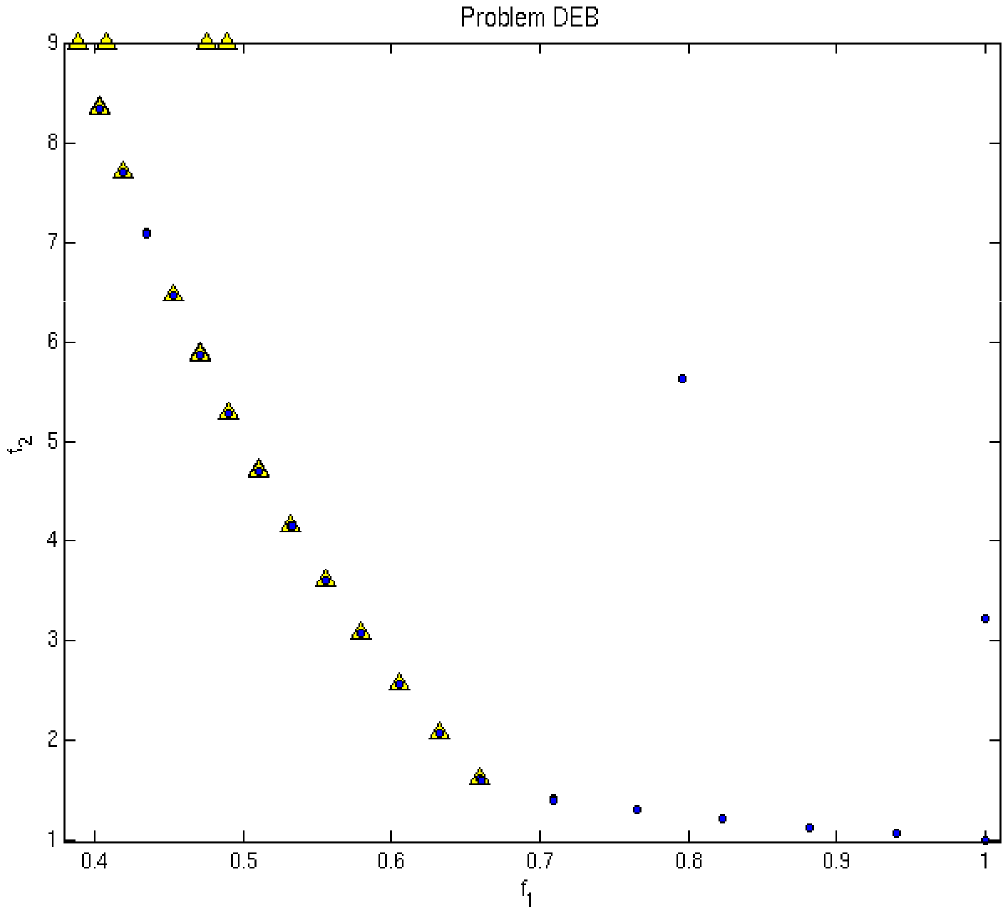
$$x_1 \in [0.1, 1]\ , \quad x_2 \in [0, 5]$$

$$
\begin{aligned}
g_1 &= x_2 + 9x_1 \geq 6 \\
g_2 &= -x_2 + 9x_1 \geq 1
\end{aligned}
$$



NLPQLP-NBI Example

Example: Problem DEB



20 NBI-subproblems
N_{des}=167

NBI-NLPQLP

Note:

since the single-objective solver of the NBI-NLPQLP scheduler is a gradient based method, NBI-NLPQLP is an accurate and fast converging algorithm.

The drawback is the low robustness of the algorithm, as expected for all the classical (gradient-based) methods.

The problem to be solved has to be smooth and well scaled.

Furthermore the Pareto curve has to be sufficiently regular.

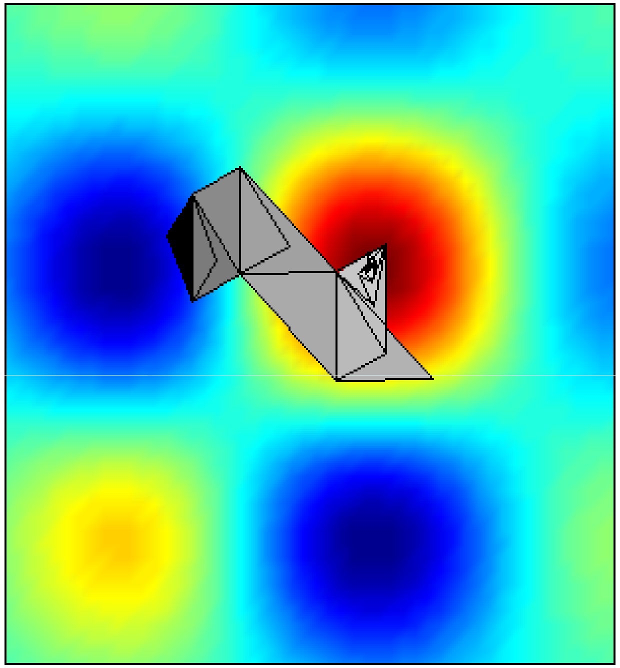
Nelder-Mead SIMPLEX - method

- **SIMPLEX** is a geometric figure with n+1 vertices in an n-dimensional space (e.g. in a 2D space is a **triangle**, in a 3D space is a **tetrahedral**)

- It does **not use the gradient** of the function (robust algorithm)

- Minimization of the target function is achieved using heuristic operators: **reflection, reflection and expansion, contraction.**

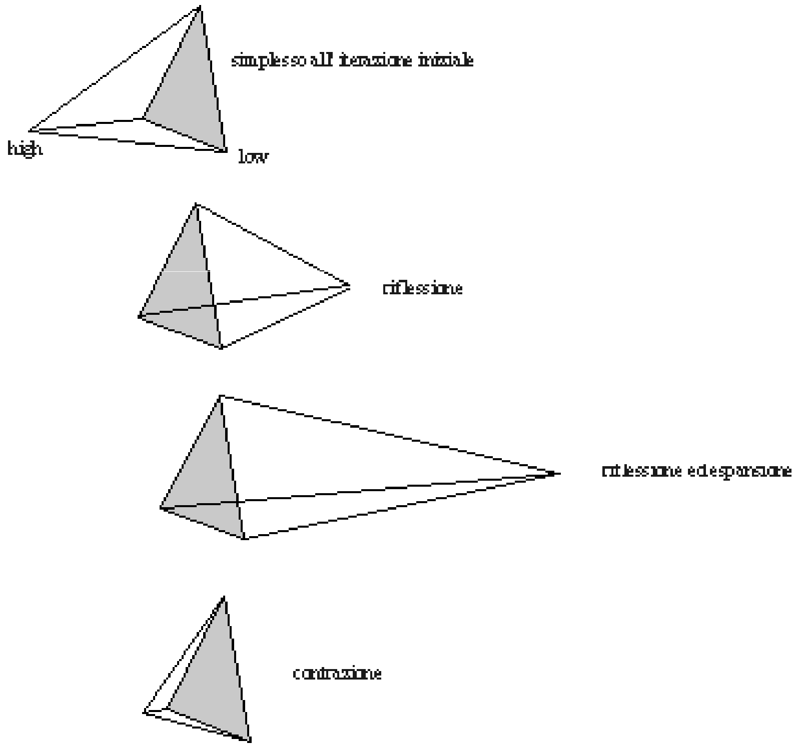
SIMPLEX



Example: hills problem

$$\max f = \exp\left(-(x^2 + y^2)/10\right)\left(\sin(x + y) + cos(x - y)\right)$$
$$x,\ y\ \in [-\pi, \pi]$$

SIMPLEX



Reflection

$$x_r = (1+\alpha)x_0 - \alpha x_l$$

$$x_0 = \frac{1}{n}\sum_{i=1, i\neq h}^{n} x_i \qquad \text{centroid}$$

$$\alpha = \frac{\|x \quad - x_0\|}{\|x_h - x_0\|} \qquad \alpha > 0$$

Expansion

$$x_e = \gamma x_r + (1-\gamma)x_0$$

$$\alpha = \frac{\|x_l - x_0\|}{\|x_r - x_0\|}$$

Contraction

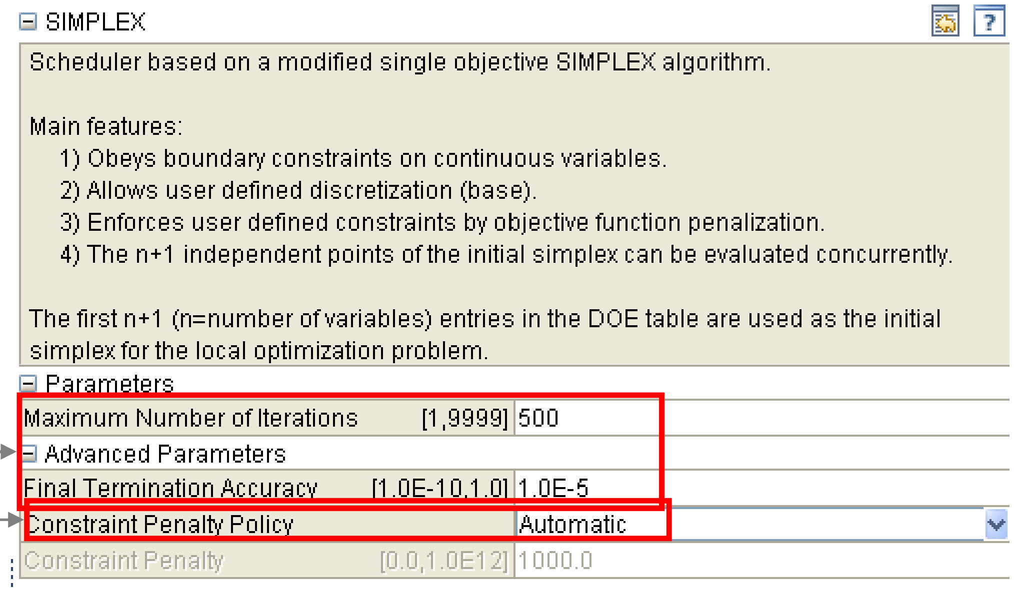
$$x_c = \beta x_h + (1-\beta)x_0$$

$$\beta = \frac{\|x_l - x_0\|}{\|x_h - x_0\|}$$

SIMPLEX Panel

Final Termination Parameters:

- Max. Num. Iterations
- Final Accuracy



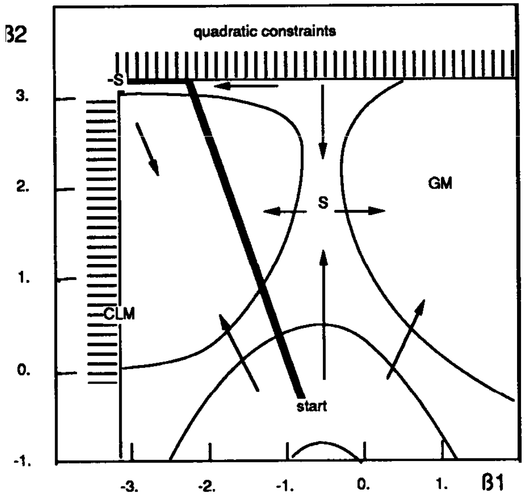
Constraint Penalty

- Automatic (Recommended)
- User Defined

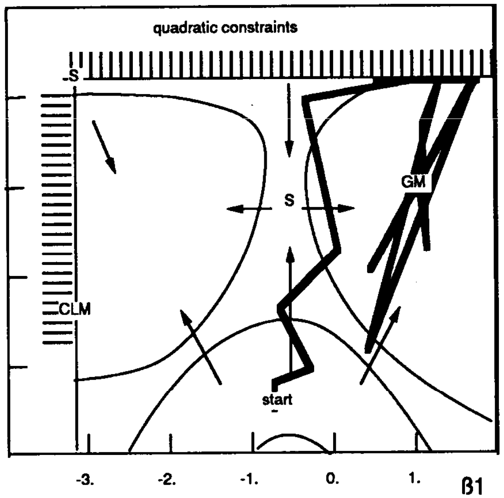
Comparison between BFGS and Simplex

Function to optimise : TEST 1

Simplex



BFGS



BFGS do not find the absolute MAX

Evolutionary Strategies

Definition

"In computer science, evolution strategy (ES) is an optimization technique based on ideas of **adaptation and evolution**. It belongs to a more general class of evolutionary computation"

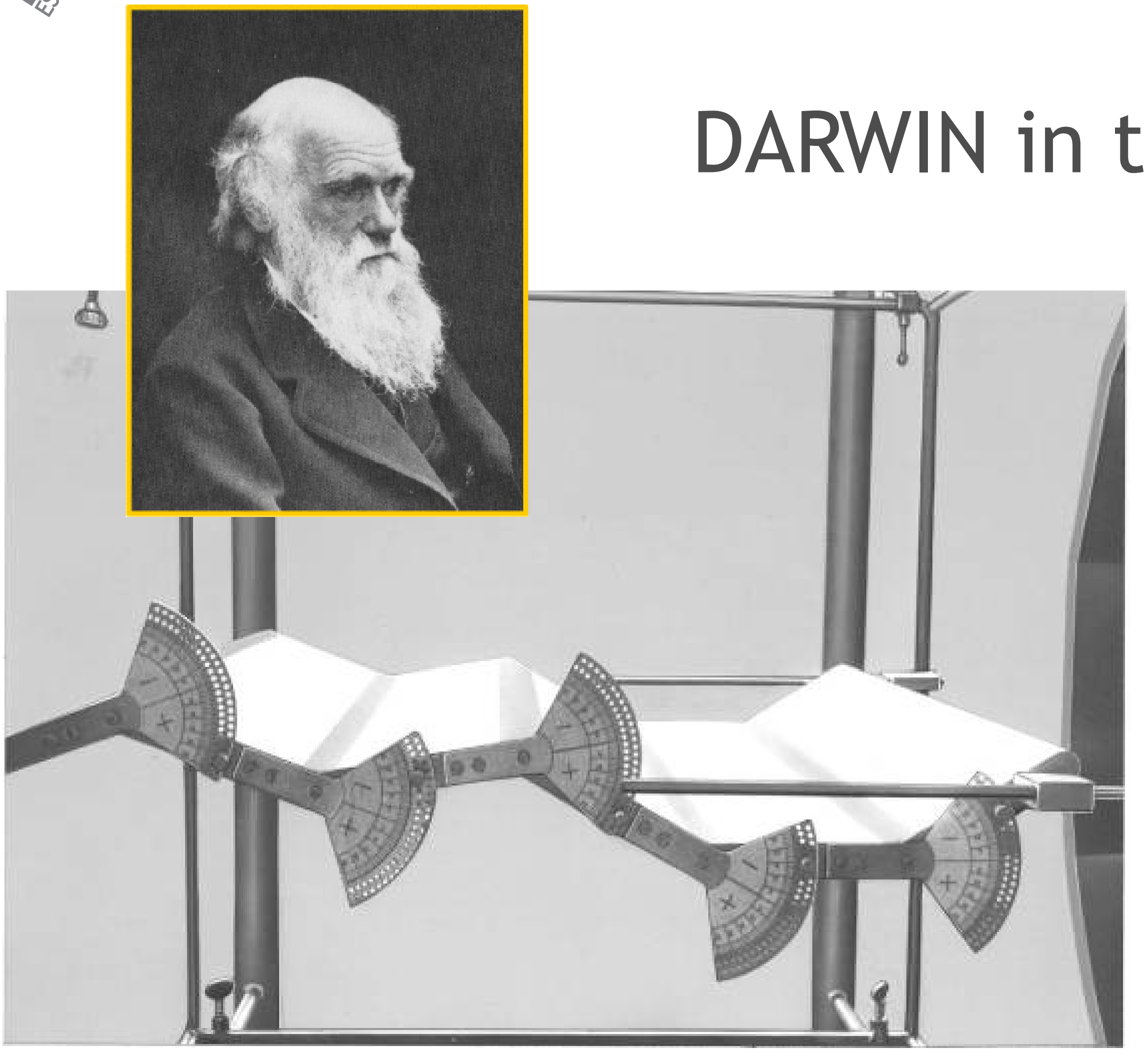
(Wikipedia – April 2007)

History

- Evolution Strategy were first used at the Technical University of Berlin
- During the search for the optimal shapes of bodies in a flow, the classical attempts with the coordinate and the well-known gradient-based strategies were unsuccessful
- The idea was conceived of proceeding strategically
- **Ingo Rechenberg** and Schwefel proposed the idea of trying random changes in the parameters defining the shape, following the example of **natural mutations**
- Thus, ES were invented to solve technical optimization problems where no analytical objective functions are usually available

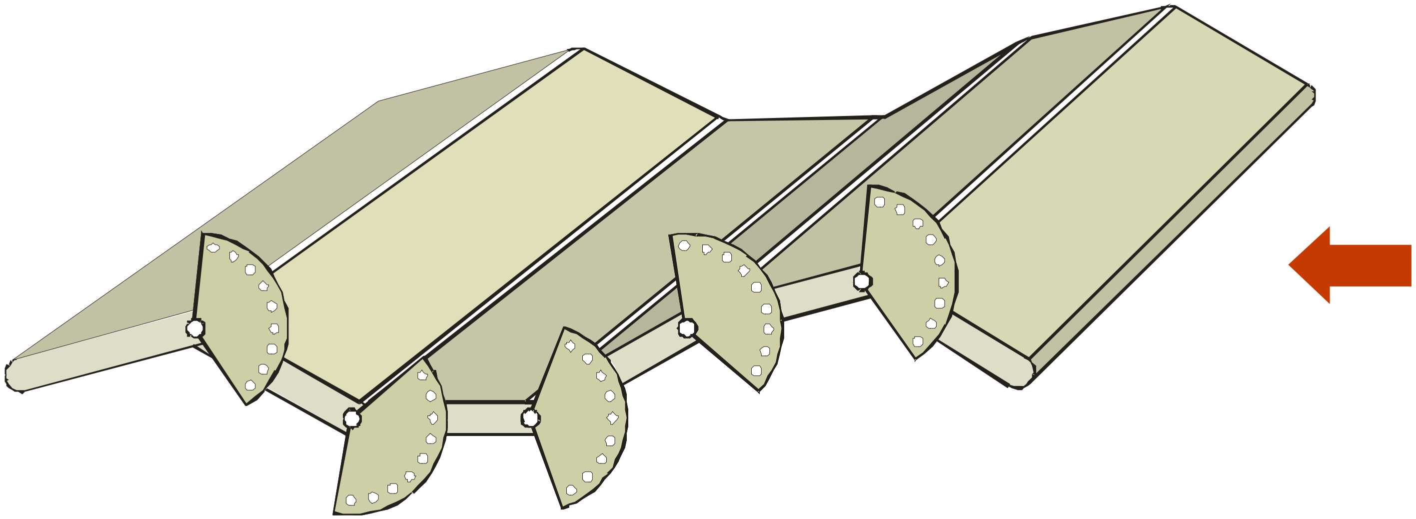
History

DARWIN in the wind tunnel!



The first real-case application of Evolution Strategy methods used by Prof. Rechenberg

The first experiment



Number of possible adjustments

$$51^5 = 345\ 025\ 251$$

History in a Magazine

Werkes — Proust an eine Bekannte: „Odette de Crécy (eine seiner Romanfiguren) ist nicht nur nicht Sie, sondern genau das Gegenteil von Ihnen" — sowie vom qualvollen Kampf um die endgültige Edition der „Recherche", den Proust unter Schmerzen und Atemnot vom Bett aus führen mußte, arbeitsfähig nur durch Koffeintabletten, getrieben von Todesangst und von Furcht, die Veröffentlichung des Werkes nicht mehr zu erleben oder den „endlosen Wälzer" (Proust) überhaupt nicht vollenden zu können.

Als sich nach Erscheinen der ersten Bände unerwarteter Erfolg einstellte — für den zweiten Band erhielt Proust 1919 den Prix Goncourt —, lebte der Moribunde für kurze Zeit noch einmal auf. Er ging wieder aus, meistens nachts, und dinierte im Hotel Ritz, wo er totenbleich, mit fiebrigen Augen erschien, in einem hocheleganten, aber derangierten Abendanzug, aus dessen Jackett wär-

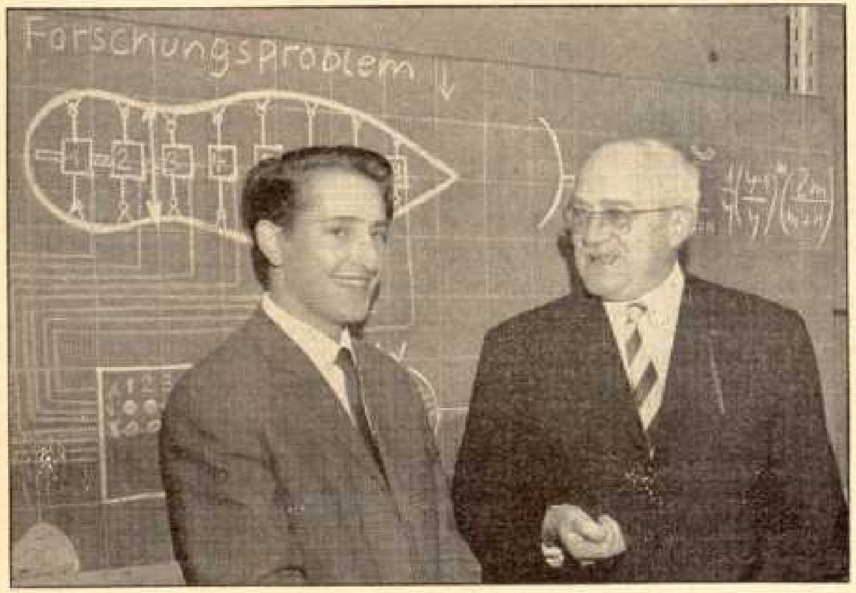
FORSCHUNG

AERODYNAMIK

Zickzack nach Darwin

Der Eingebung und oftmals auch glücklichem Zufall verdanken Generationen von Flugzeugtechnikern zukunftweisende Lösungen. Aber ein Student der Technischen Universität in West-Berlin möchte den Fortschritt kalkulabel machen: Er fand für das Roulette-Spiel der Flugzeugingenieure ein System.

Zahllose aufwendige Versuchsreihen in mietshausgroßen Windkanälen, deren Bau Millionen Dollar kostet und in denen Mammut-Flügelräder leichte Brise ebenso wie heulenden Orkan oder mehrfach schallschnelle Luftströme erzeugen können, sind bei den großen Flugzeugfirmen nötig, um für ein neues



Student **Rechenberg**, Lehrer **Wille***: Roulette in der Hochschule

DER SPIEGEL
18th November 1964

Zigzag after DARWIN

Biological Evolution

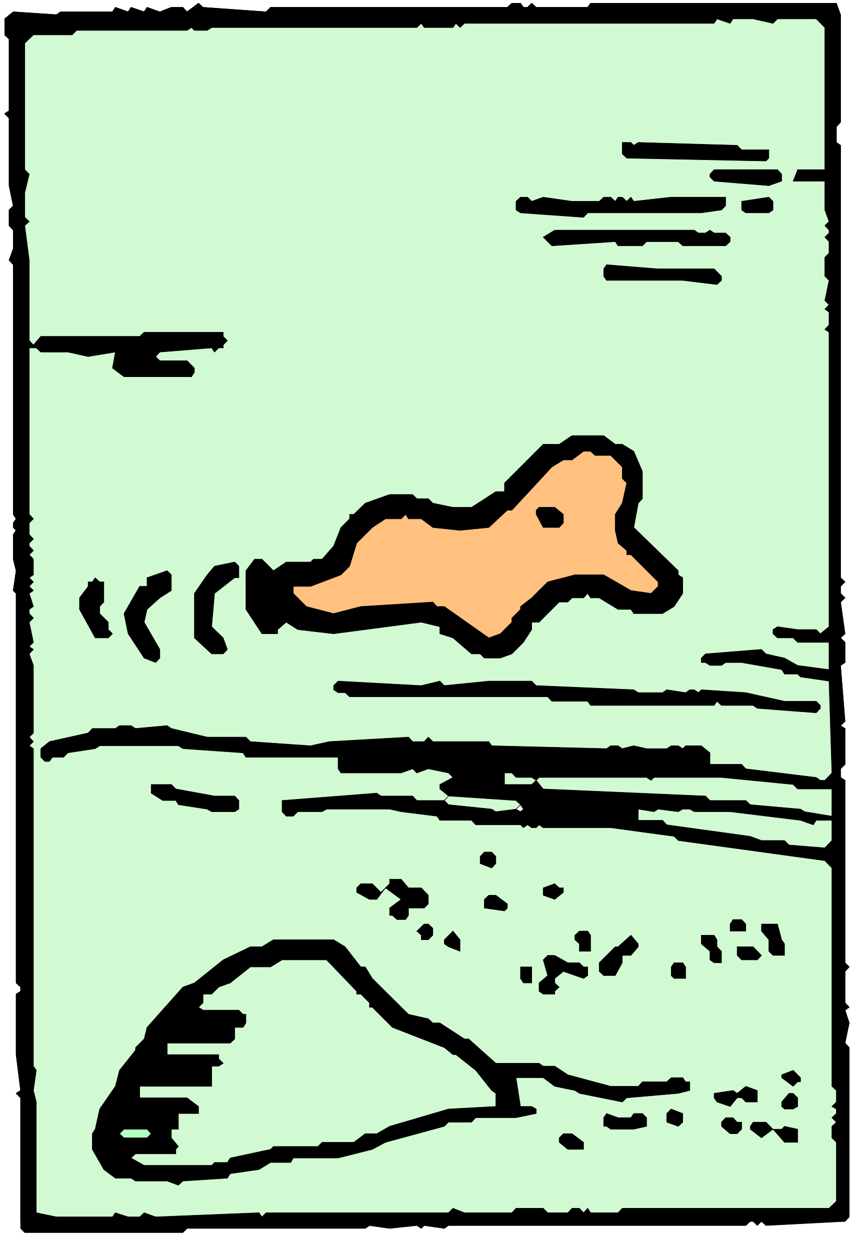
ES belongs to the family of evolutionary algorithms (EAs) together with Genetic Algorithms (GAs).

EAs are direct global search methods based the model of organic evolution.

Nature's Way of Optimization

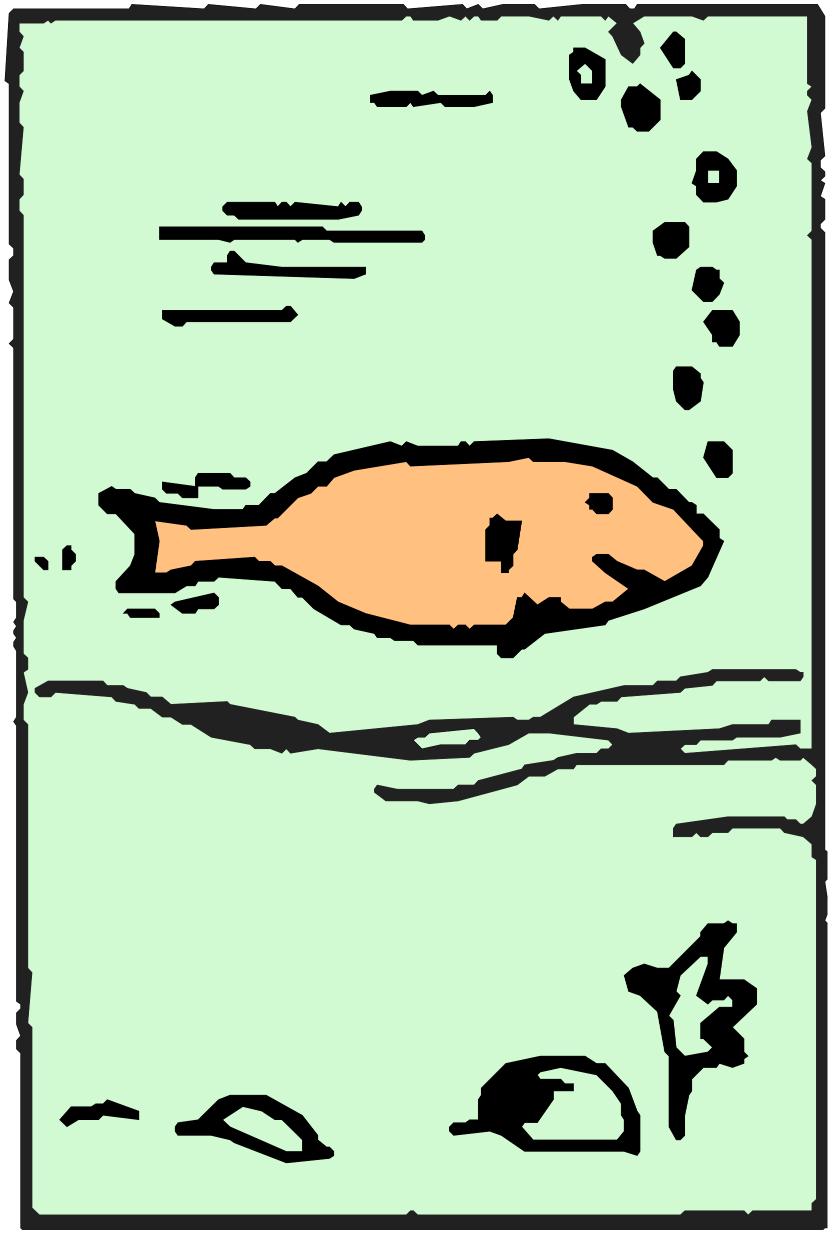
1

Protoplasm



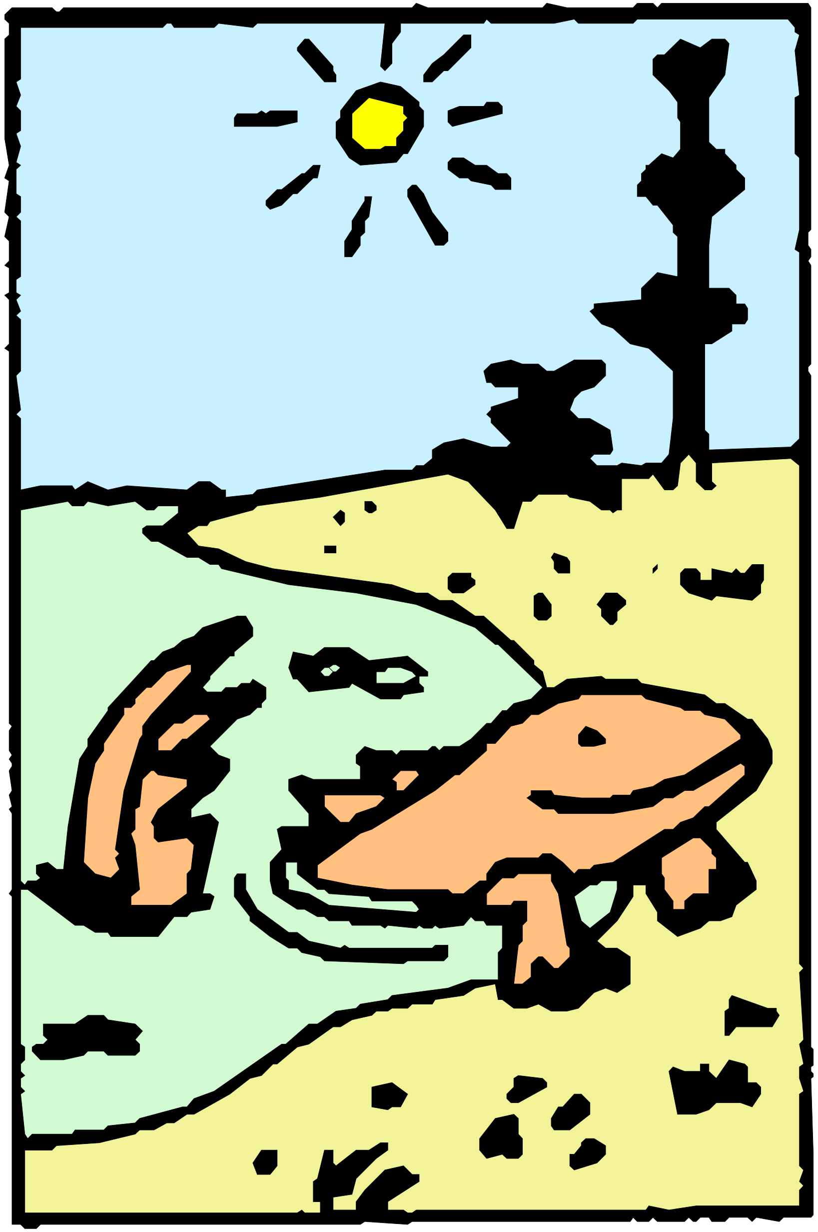
2

fish



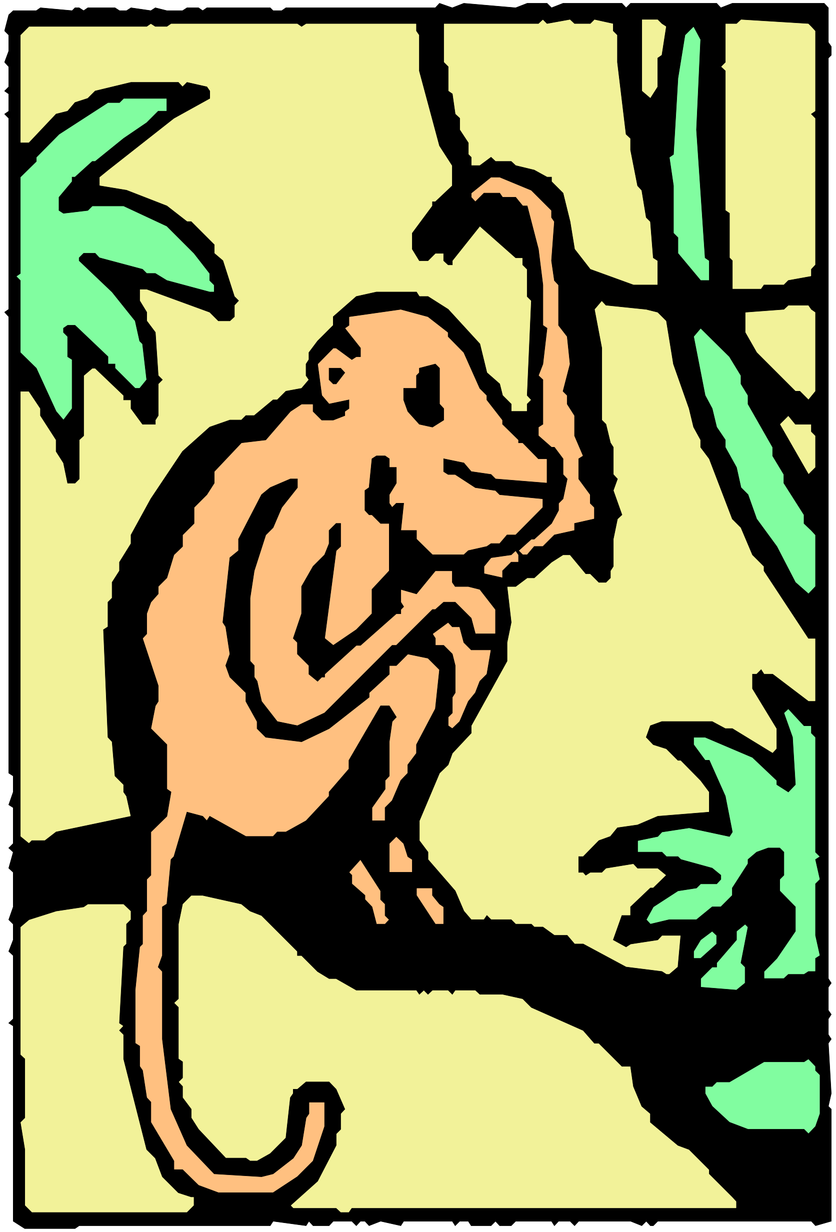
3

Life peeks out of the water and spreads over the country



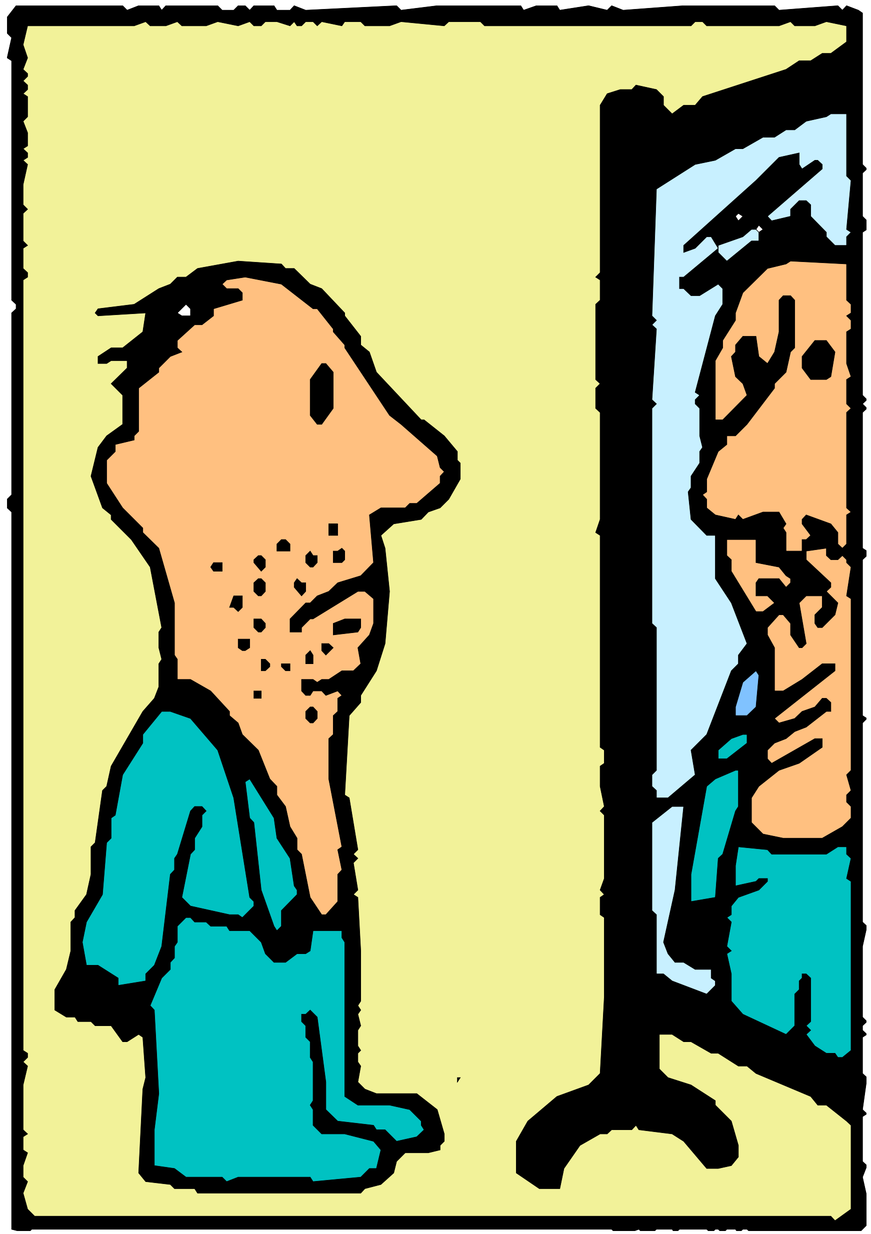
4

Our ancestors climb
the treetops



5

And finally... Human being



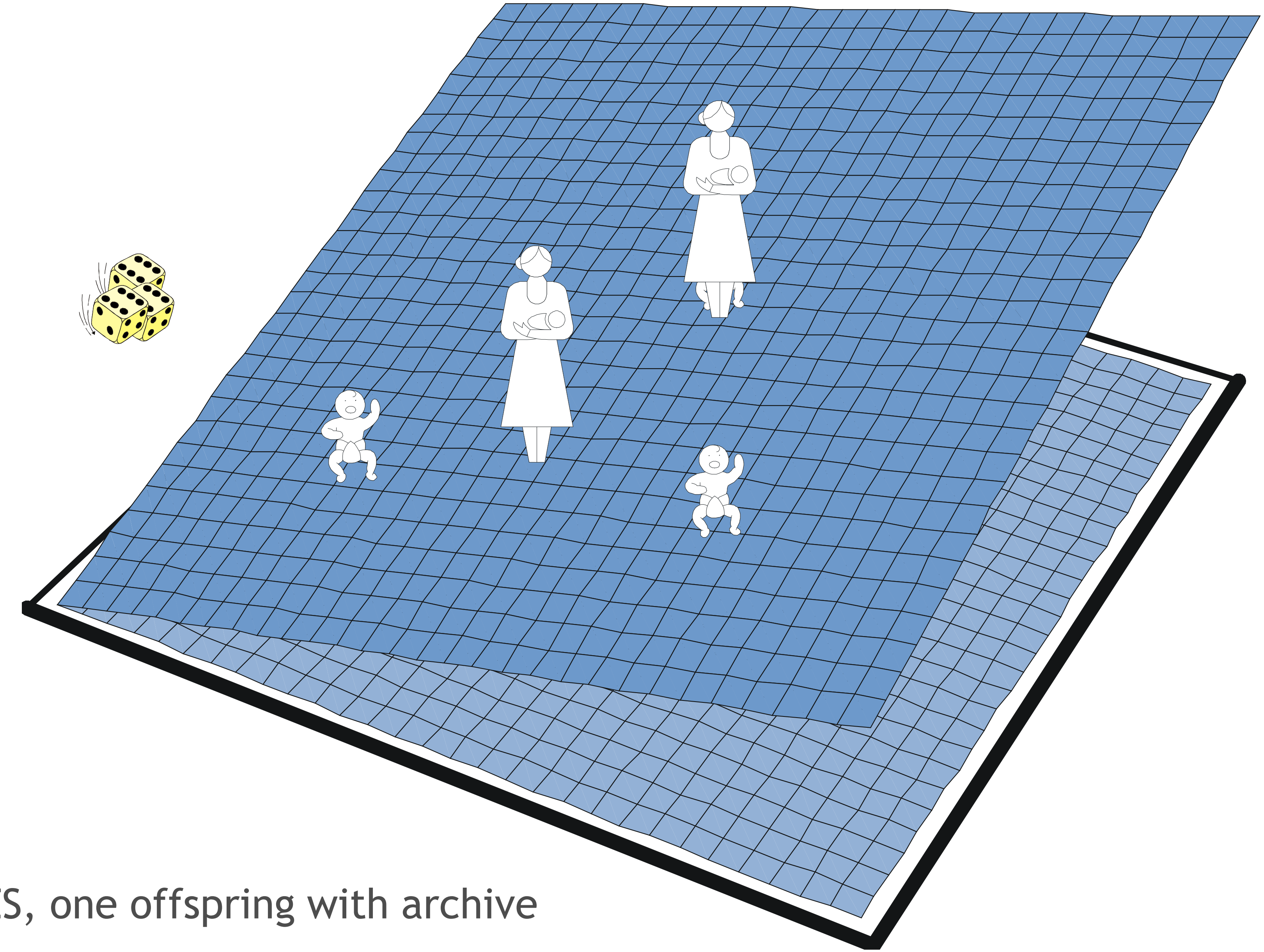
Glossary: Terms and Operators

- Terms:
  - **Gene**: design parameter
  - **Individual**: design
  - **Elite**: preferred design
  - **Fitness**: objective function
  - **Population**: set of individuals
  - **Archive**: set of saved good individuals
- Operators:
  - **Selection**
  - **Reproduction**
  - **Mutation**

Evolutionary Strategy General Scheme

1. Initial population
2. Evaluate individuals
3. **Selection** of the best individual(s)
4. **Recombination**
5. **Mutation**
6. Evaluate individuals
7. Return to step 3 until convergence is reached
8. End

(1 + 1)-ES



(1+1)-ES, one offspring with archive

(1 + 1)-ES in modeFRONTIER 4

⊟ Evolution Strategy

Scheduler based on (mu+lambda)-Evolution Strategy and (mu,lambda)-Evolution Strategy.

Main features are:
1) Self-adaptive refinement of step-sizes.
2) Handles user defined constraints by means of Pareto ranking.
3) Supports mixed discrete and continuous optimization.
4) Diversity and spread of solutions is guaranteed without use of sharing parameters.
5) Allows concurrent evaluation of independent individuals.

The first (mu) entries in the DOE table are used as the problem's initial population.

⊟ Parameters

Number of Generations	[1,5000]	100
Number of Offsprings	[1,100]	1

⊟ Advanced Parameters

Initial Stepsize (% of Range)	[0.0,1.0]	0.1
Minimal Stepsize (% of Range)	[0.0,1.0]	0.01
Selection Type		+
Recombination		Off
Random Generator Seed	[0,999]	1

⊟ Category Parameters

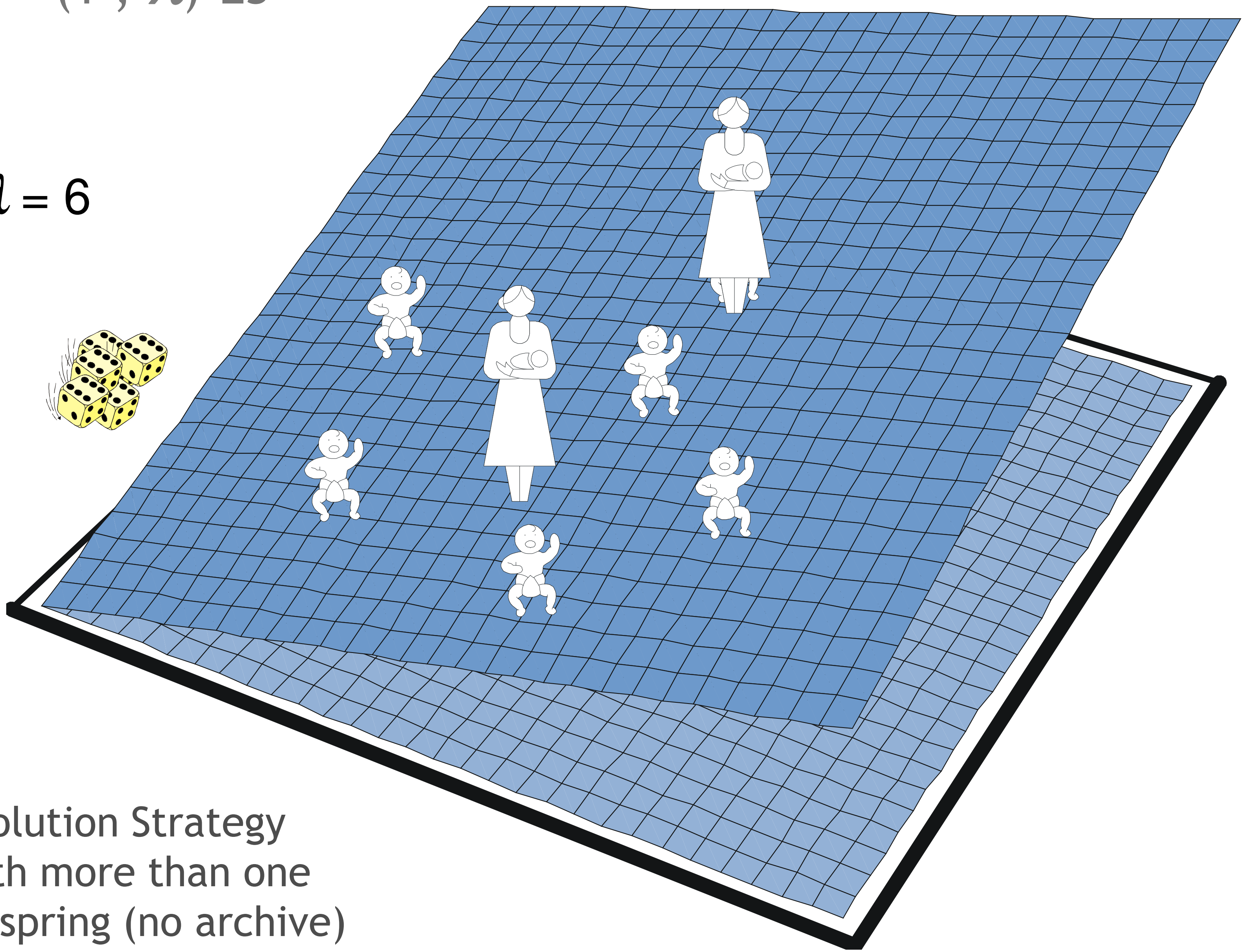
Categorize Generations	☐

To set (1+1)-ES:

1. Set 1 point in the DOE table
2. Set the number of offsprings equal to 1
3. Set the selection type equal to +
4. Switch off the recombination

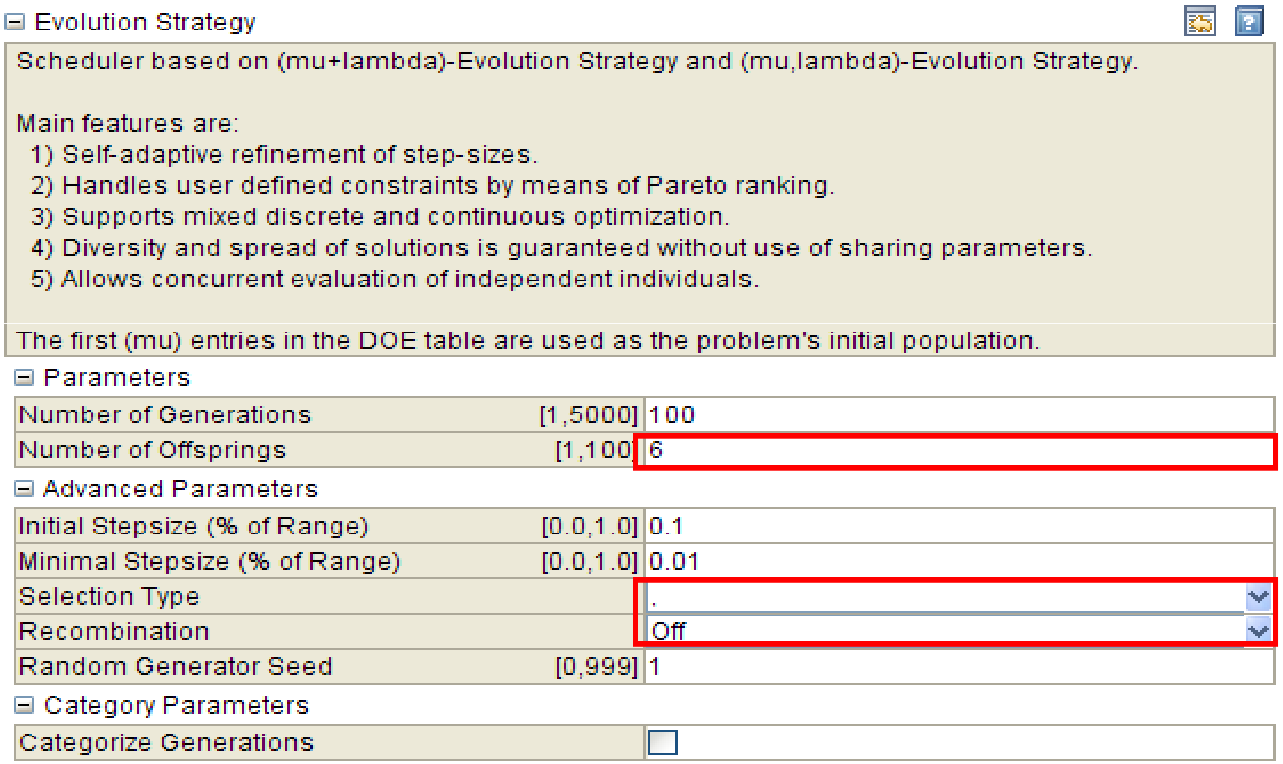
(1 , λ)-ES

$\lambda = 6$



Evolution Strategy
with more than one
offspring (no archive)

(1 , 6)- ES in modeFRONTIER 4



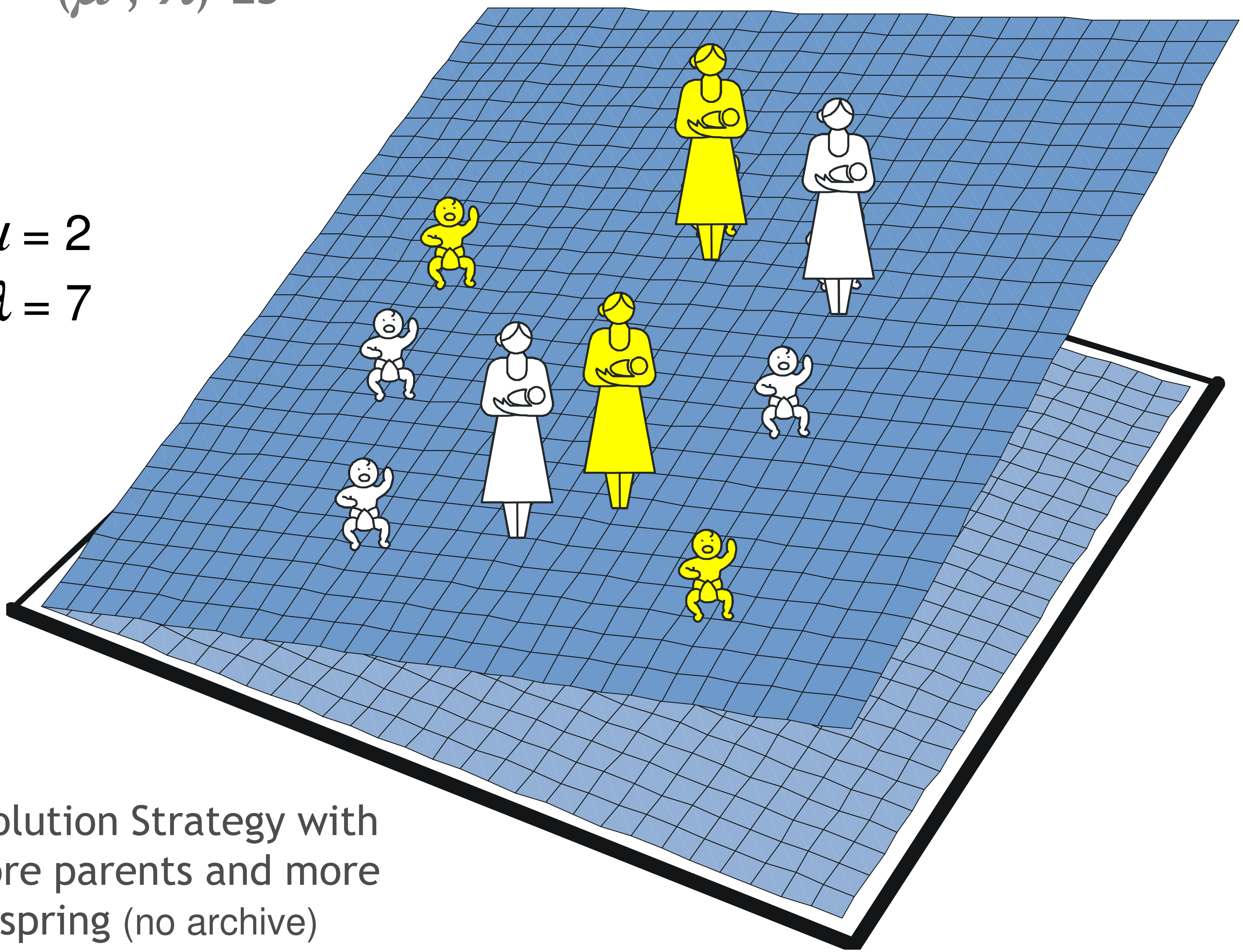
To set (1,6)-ES:

1. Set 1 point in the DOE table
2. Set the number of offsprings equal to 6
3. Set the selection type equal to ,
4. Switch off the recombination

(μ , λ)-ES

$\mu = 2$

$\lambda = 7$



Evolution Strategy with more parents and more offspring (no archive)

(2 , 7)- ES in modeFRONTIER 4

Evolution Strategy

Scheduler based on (mu+lambda)-Evolution Strategy and (mu,lambda)-Evolution Strategy.

Main features are:
1) Self-adaptive refinement of step-sizes.
2) Handles user defined constraints by means of Pareto ranking.
3) Supports mixed discrete and continuous optimization.
4) Diversity and spread of solutions is guaranteed without use of sharing parameters.
5) Allows concurrent evaluation of independent individuals.

The first (mu) entries in the DOE table are used as the problem's initial population.

Parameters

Number of Generations	[1,5000]	100
Number of Offsprings	[1,100]	7

Advanced Parameters

Initial Stepsize (% of Range)	[0.0,1.0]	0.1
Minimal Stepsize (% of Range)	[0.0,1.0]	0.01
Selection Type		,
Recombination		Off
Random Generator Seed	[0,999]	1

Category Parameters

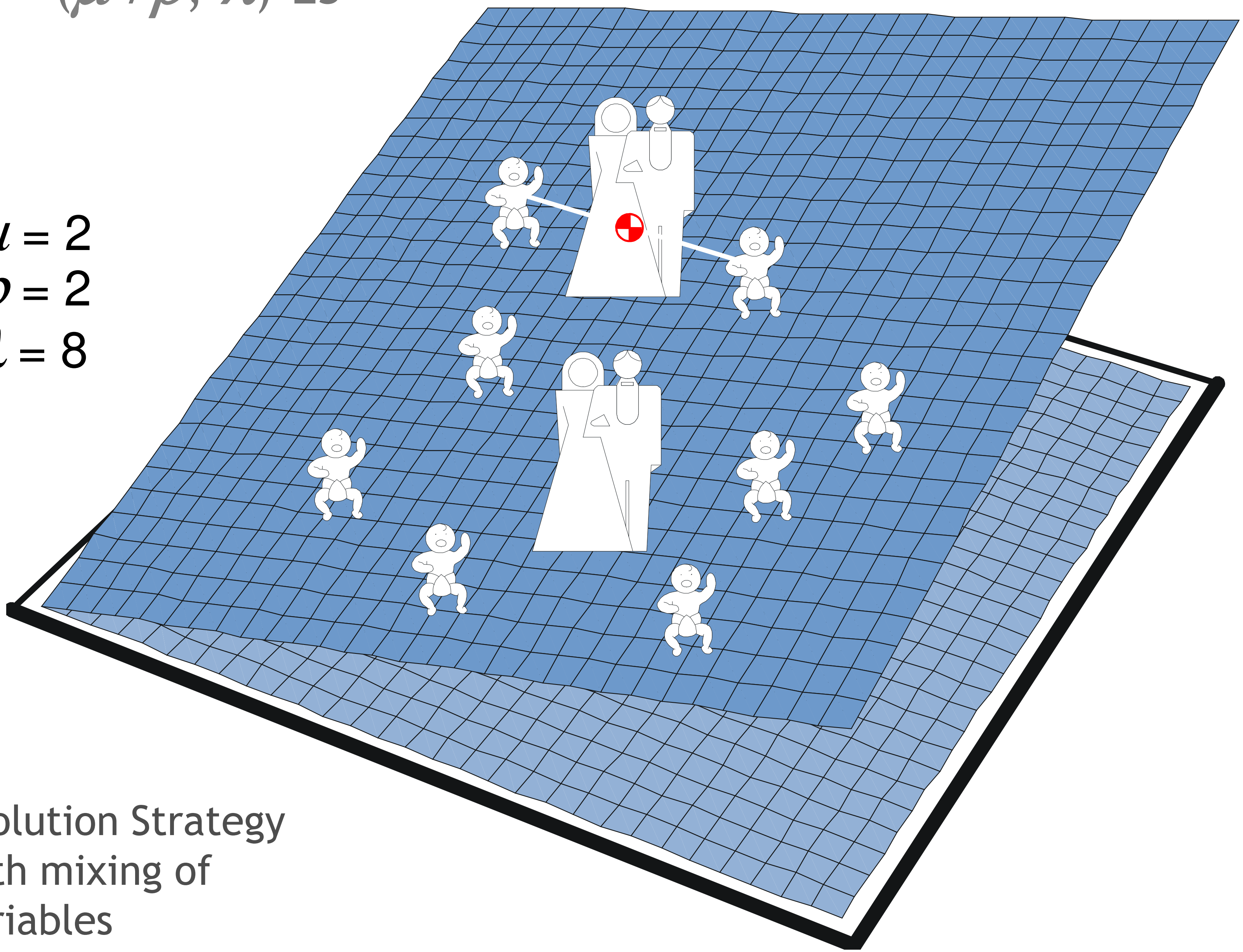
Categorize Generations	☐

To set (2,7)-ES:

1. Set 2 point in the DOE table
2. Set the number of offsprings equal to 7
3. Set the selection type equal to ,
4. Switch off the recombination

$(\mu / \rho, \lambda)$-ES

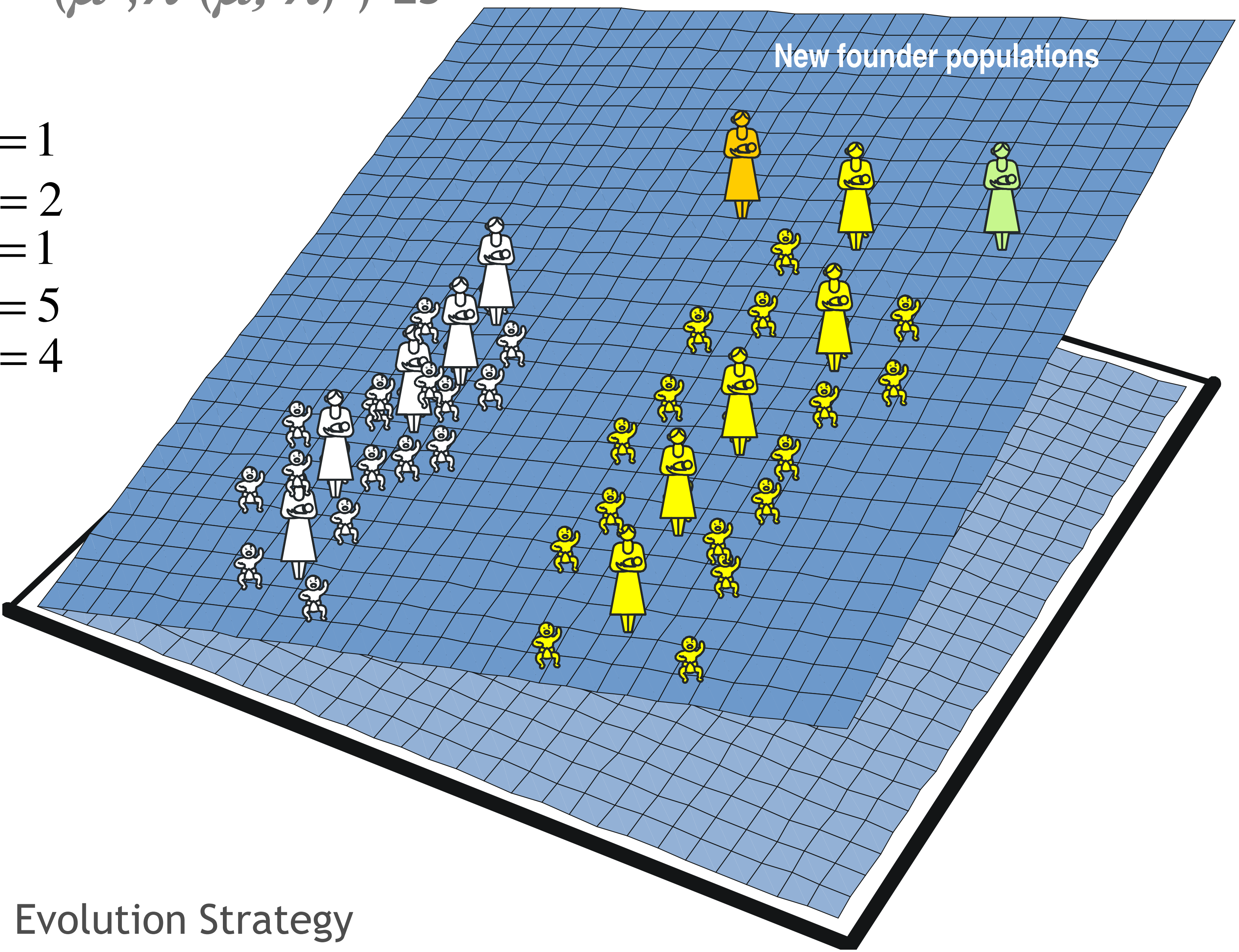
$\mu = 2$
$\rho = 2$
$\lambda = 8$



Evolution Strategy with mixing of variables

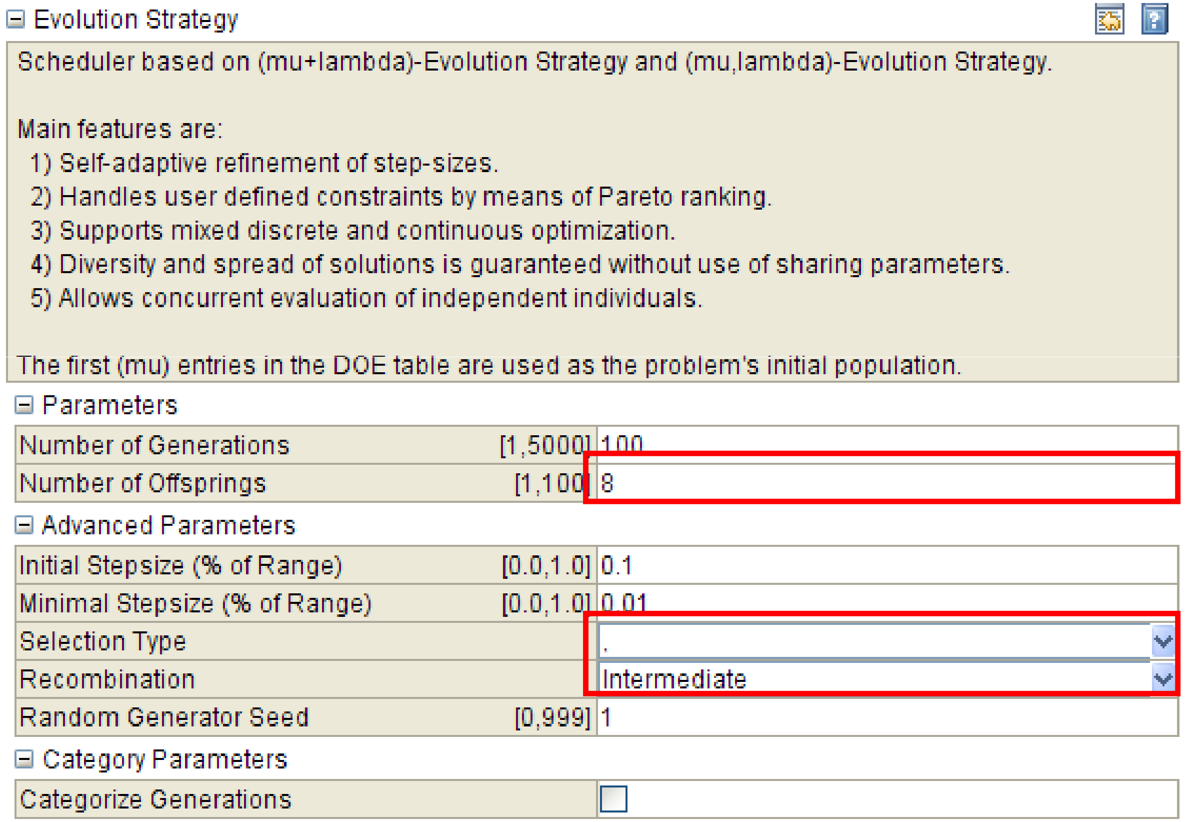
$(\mu', \lambda'(\mu, \lambda)^{\gamma})$-ES

$$\mu' = 1$$
$$\lambda' = 2$$
$$\mu = 1$$
$$\lambda = 5$$
$$\gamma = 4$$



Nested Evolution Strategy

(2 /2, 8)- ES in modeFRONTIER 4



To set (2/2,8)-ES:

1. Set 2 point in the DOE table
2. Set the number of offsprings equal to 8
3. Set the selection type equal to ,
4. Switch on the intermediate recombination

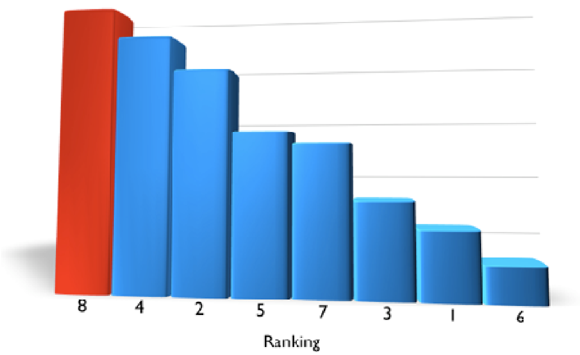
ES notations

- Different notations specify how the population of the next generation is formed out of a set of best individuals of the old population
- Different strategies:
  - (μ , λ)-ES, the archive size is equal to zero. The μ best out of λ offspring completly replace the old population
  - (μ + λ)-ES, the archive size is equal to μ (this represents a kind of **elitism**). The μ best out of λ offspring and μ from the old pupulation are selected
  - (μ/ρ + λ)-ES, the archive size is equal to μ/ρ. Only μ/ρ best individual contribute to bild up the offspring

Parent selection

- Selection plays a crucial role on performance and convergence
- Using ***plus*-selection** (archive size greater than zero) is usually recommended for combinatorial optimization problems

- Parents can be selected in different ways:
  - every individual has the same probability to be selected
  - with a probability that is **proportional to the fitness**



Survivor selection

- Another kind of selection is applied after creating λ children from the μ parents by mutation and recombination

- Deterministically **chops off the bad results**

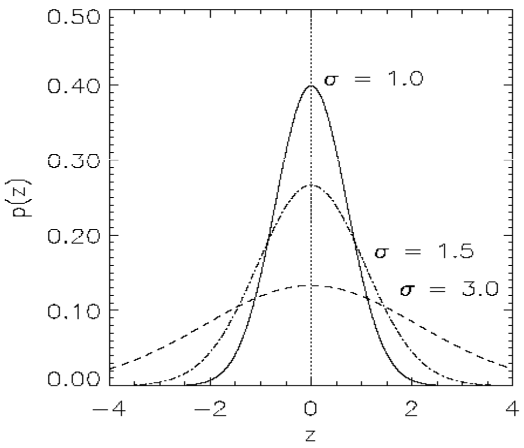
- Selection of the **best results** basis either on:
    - The set of children only: (μ,λ)-ES. This selection can "forget" some good results
    - The set of parents and children: $(\mu+\lambda)$-ES. This selection is a kind of elitist strategy

Mutation and Recombination

- Mutation and recombination operators in ESs are problem-specifically designed
- Depending on the search space and objective function, the recombination and/or the mutation of the strategy parameters may occur differently
- Mutation represents the main source of variation
- Recombination is applied whenever possible and useful. It uses ρ or more parental individuals to generate a single recombinant.

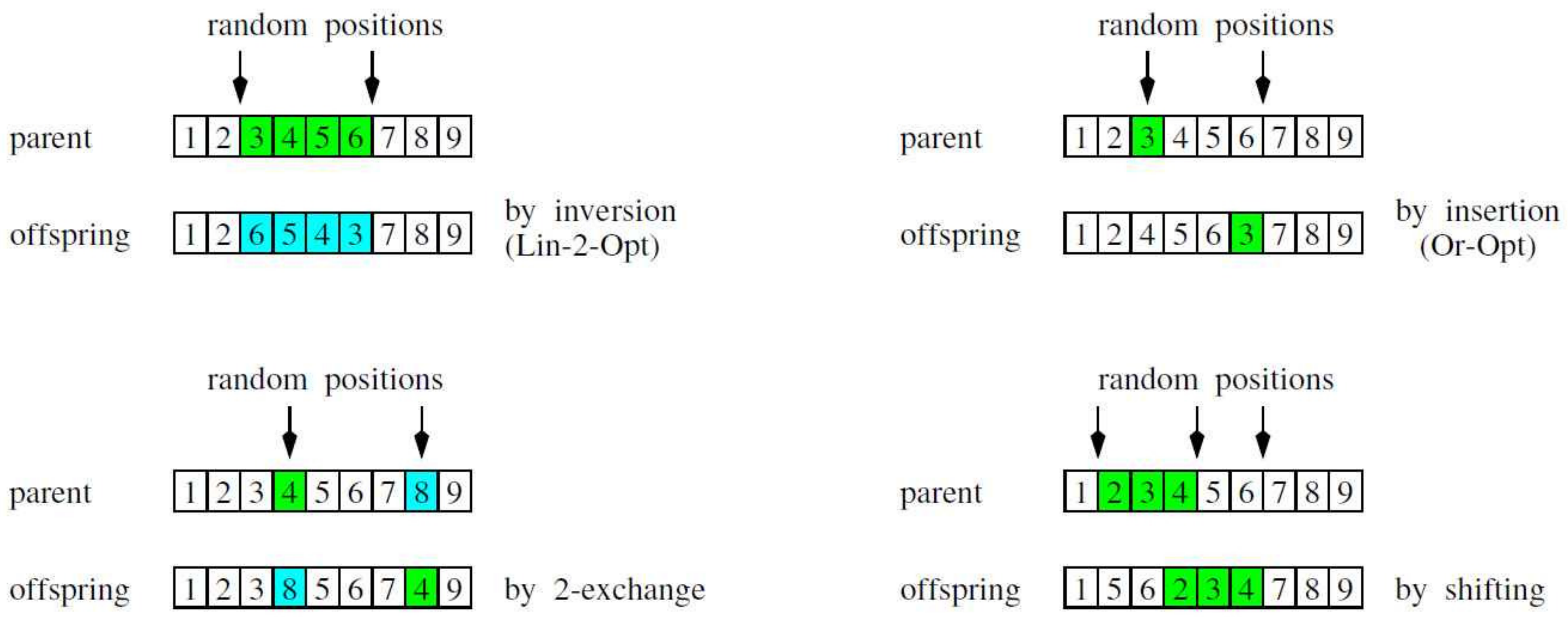
Mutation with continuous variables

- Mutation is based on a normal distribution, a random variation of the genes
- The standard deviation of the normal distribution **changes during the generations (adaptation)**
- Usually the standard deviation decreases continuously
- Each design variable is assigned a standard deviation for generating an appropriate mutation step
- In **CMA**-ES the shape of mutation distribution is generated according to a covariance matrix C that is adapted during evolution



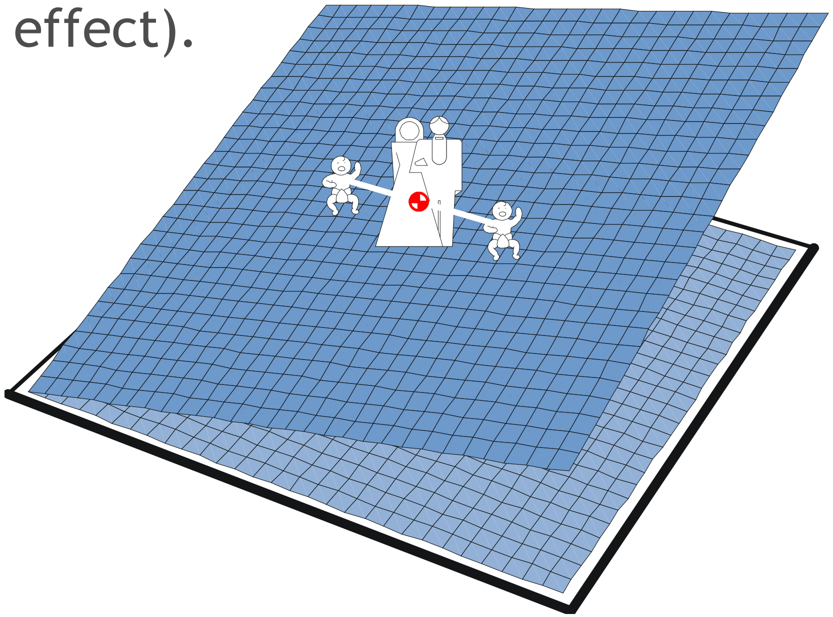
Mutation with discrete variables

Examples of mutation with discrete variables:



Recombination

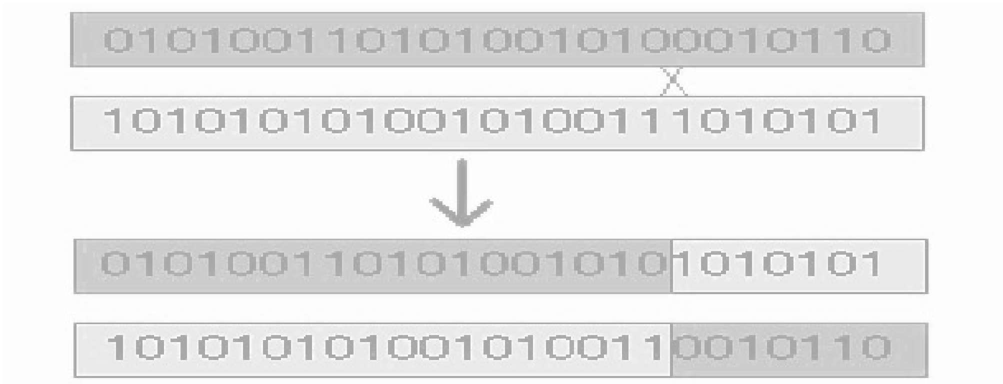
- The main goal of recombination is the conservation of common components of the parents
- Recombination transfers the beneficial similarities to the next generation
- Recombination damps the effect of malicious components of the parents' genes (genetic repair effect).



Recombination using continuous variables

Recombination with discrete variables

- One-point crossover is the most classical operator for recombination
- Two parents are chosen and some portion of the genetic material is exchanged between the parent variables vectors.
- The point of the crossing site is randomly chosen and the binary strings are cut at that point.
- The two heads are then swapped and rejoined with the two tails. From the resulting individuals one is randomly selected to be the new individual.



How to select the best strategy?

which the reverse is true?" To address t[illegible] on we compare the sum over all f of $P(d^y_m|f,m,a_1)$ to the sum over all f of P[illegible]$a_2)$. This comparison constitutes a major result of this paper: $P(d^y_m|f,m,a)$ is in[illegible] of a when we average over all cost functions:

Theorem 1 *For any pair of algorith[illegible] nd* a_2,

$$\sum_f P(d^y_m|f \ldots = \sum_f P(d^y_m|f,m,a_2).$$

A proof of this result is found in Appe[illegible] An immediate corollary of this result is that for [illegible] $(\Phi(d^y_m)|f,m,a)$ is independent [illegible] nance measure is unimportant.

The No Free Lunch Theorem for Optimization*

This theorem explicitly demonstrates that what an algorithm gains in performance on one class of problems it necessarily pays for on the remaining problems; that is the only way that all algorithms can have the same f-averaged performance.

*D. H. Wolpert, W.G. Macready

Hints

- Larger offspring **population sizes** (λ) take longer to converge, but can find better solutions
- Intermediate recombination on object variables helps to overcome premature convergence problems

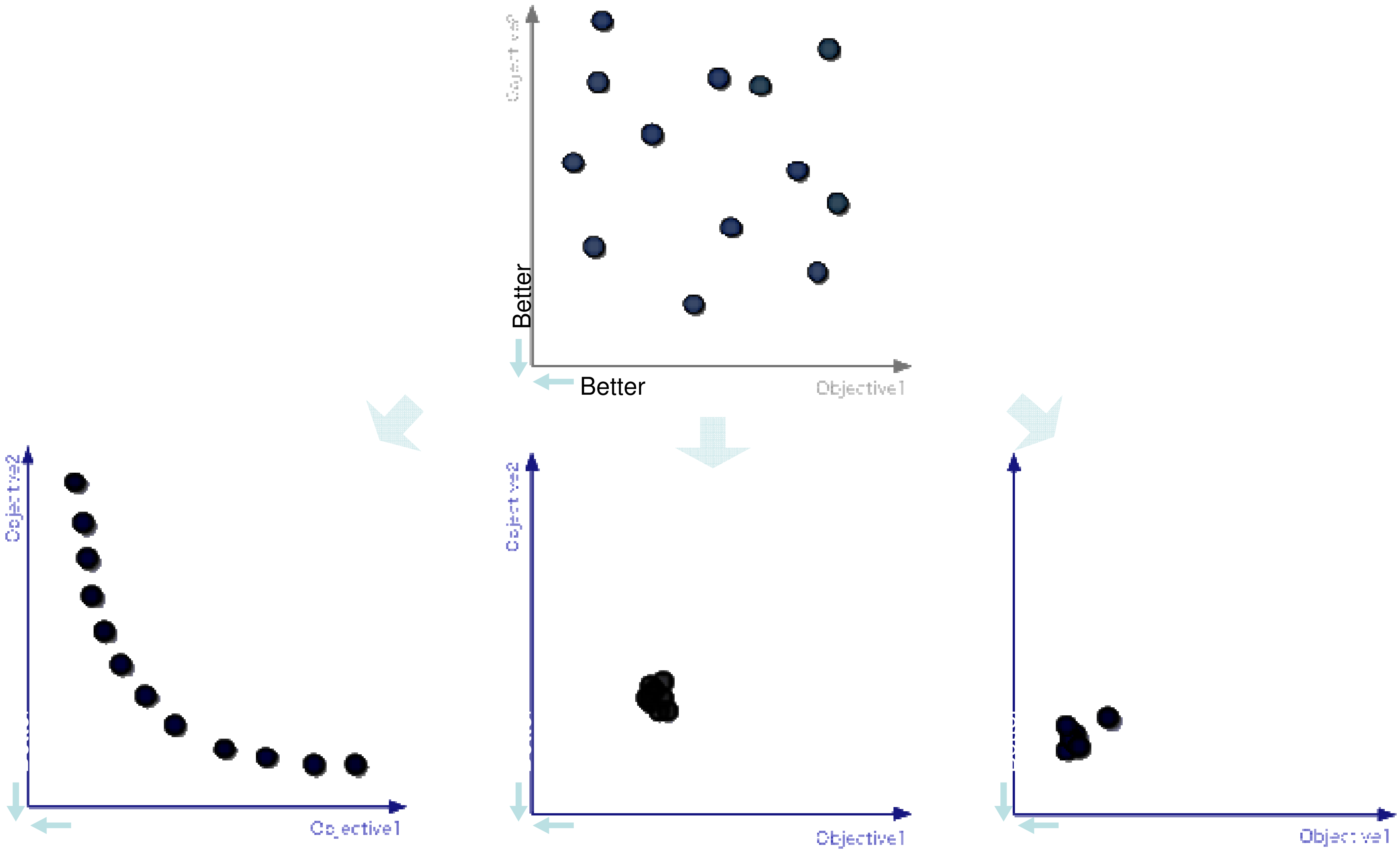
- Using **large values for initial perturbation** ones will increase the time to converge but the method results to be **more robust**
- On the contrary, using smaller ones will increase the probability of premature convergence

Application of ES

- ES methods are **recommended** for:
  - Scalable to high-dimensional optimization problems
  - Problems well-suited for evolutionary improvement of designs
  - 0/1 problems
  - Continuous, discrete and binary variables
  - Large number of constraints
- **Advantages**:
  - Always converge to a good enough solution in successive, self-contained stages
  - No gradients are necessary
  - Robustness against noisy objective functions.
  - Parallelization
- **Shortcomings**
  - Slow convergence

Few words on the convergence of multiobjective algorithms

Difference in Single Objective GA and Multi-Objective GA



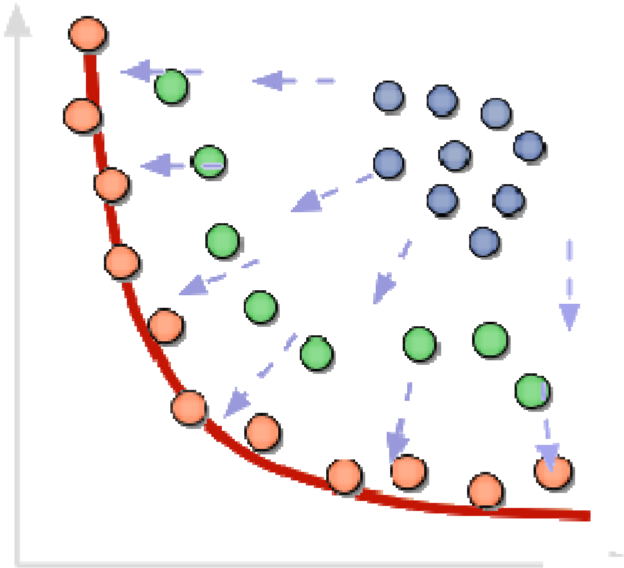
MOGA applied to trade-off problem

SOGA applied to trade-off problem

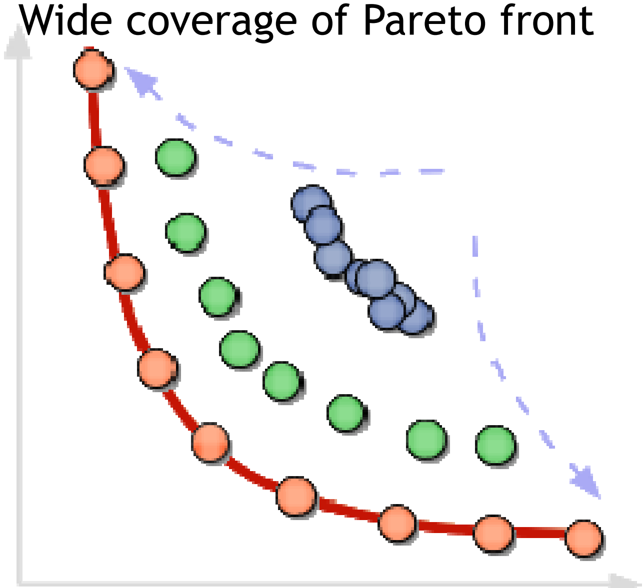
SOGA applied to MO Problem without trade-off

Desirable Features in Multi-objective

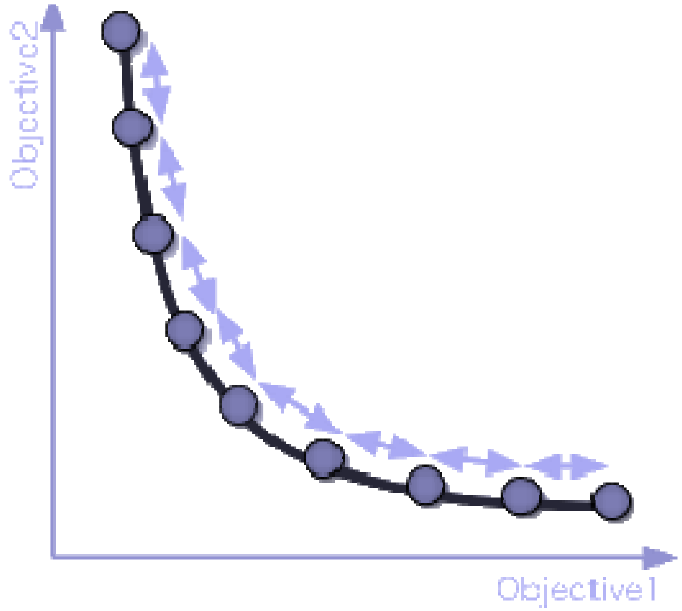
Approach to Pareto Front



Wide coverage of Pareto front



Uniform distribution on Pareto front



References

- Rechenberg, I. (1973) "Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution", Stuttgart: Fromman-Holzboog
- Hans-Paul Schwefel: Evolution and Optimum Seeking: New York: Wiley & Sons 1995.
- Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution Strategies: A Comprehensive Introduction. In *Natural Computing*, 1(1):3-52.
- EMOO Web page: http://www.lania.mx/~ccoello/EMOO