EVOLUTIONARY ALGORITHMS:

What are EAs?
Mathematical Formulation & Computer Implementation
Multi-objective Optimization, Constraints
Computing cost reduction

Kyriakos C. Giannakoglou

Associate Professor,

Lab. of Thermal Turbomachines (LTT),

National Technical University of Athens (NTUA),

GREECE

Outline

- **From traditional problem solving techniques to EAs.**
- **Generalized EA: Basic and advanced operators.**
- **Mathematical foundations of EAs.**
- **EAs for multi-objective optimization.**
- **Distributed Evolutionary Algorithms (DGAs).**
- **Hierarchical Evolutionary Algorithms (HGAs).**
- **Constraints’ handling.**
- **Efficient ways for reducing the computing cost of EAs.**
- **Applications in the field of aeronautics, turbomachinery, energy production, logistics.**

Effective/Efficient Problem Solving Techniques

- **The number of possible solutions in the search space is so large as to forbid exhaustive search**
- **Seeking the best combination of approaches that addresses the purpose to be achieved**
- **Finding the solution using the available computing resources**
- **Finding the solution within the available time**
- **One or more (contradictory) targets**
- **Solving the problem under a number of (hard/soft) constraints**

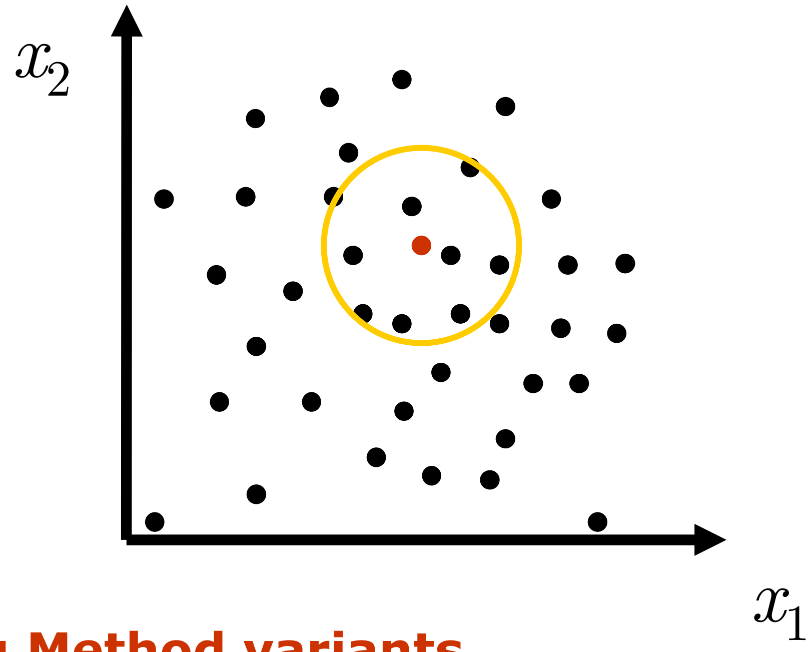
Basic Concepts of Problem Solving Techniques

- **<u>Representation:</u> how to encode alternative candidate solutions for manipulation**
- **<u>Objective:</u> describes the purpose to be fulfilled**
- **<u>Evaluation function:</u> returns a value that indicates the (numeric of ordinal) quality of any particular solution, given the representation**

Hill-Climbing: A Traditional (Deterministic) PST

Useful Definitions:

- **Neighborhood of a solution**
- **Local Optimum**



Requirements:

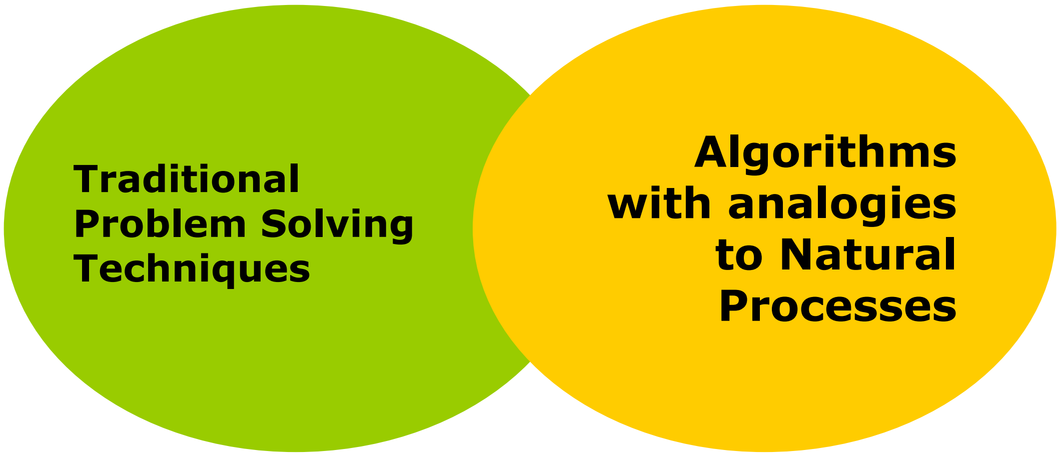
- **A starting point**
- **Computation of gradient**
- **Termination criteria**

Ideas for creating Hill-Climbing Method variants

- **How to select the new solution for comparison with the current solution (how to compute the gradient) …**
- **To use more than one starting solutions, if necessary …**
- **To be "less deterministic" …**

Algorithms Relying on Analogies to Natural Processes

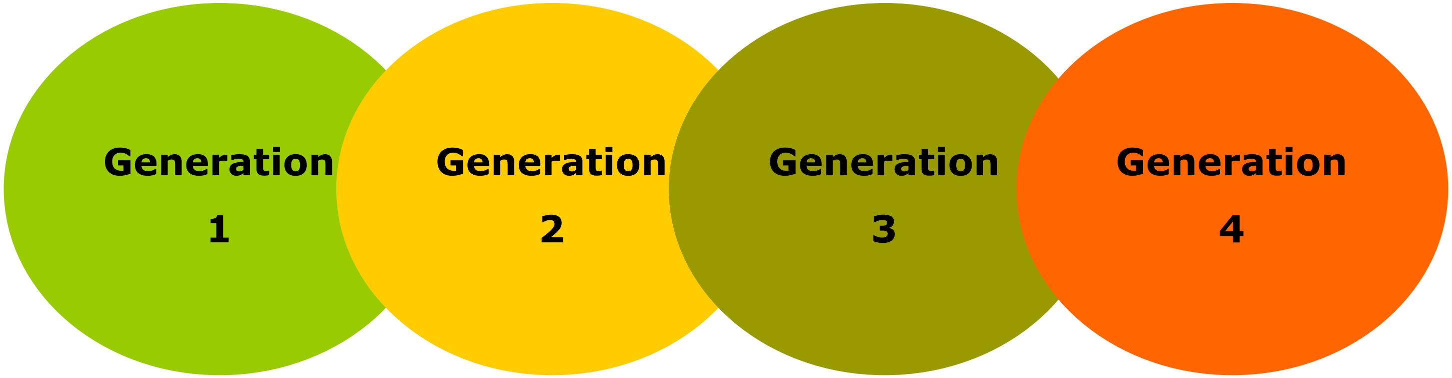
- **Evolutionary Programming**
- **Genetic Algorithms**
- **Evolution Strategies**
- **Simulated Annealing**
- **Classifier Systems**
- **Neural Networks**



The Subclass we are interested in …

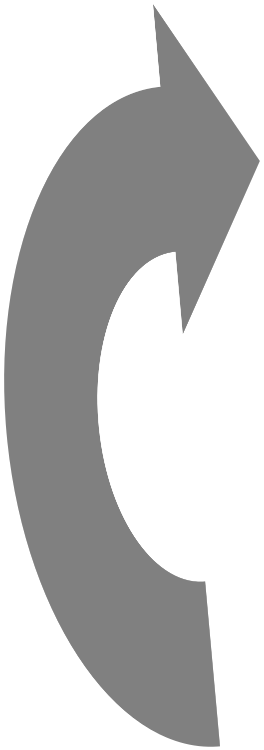
Methods which are based on the principle of evolution (i.e. the survival of the fittest)

- **Handling populations of candidate solutions**
- **Undergoing unary (mutation-type) operations**
- **Undergoing higher-order (crossover-type) operations**
- **Using a selection scheme biased towards fitter individuals**



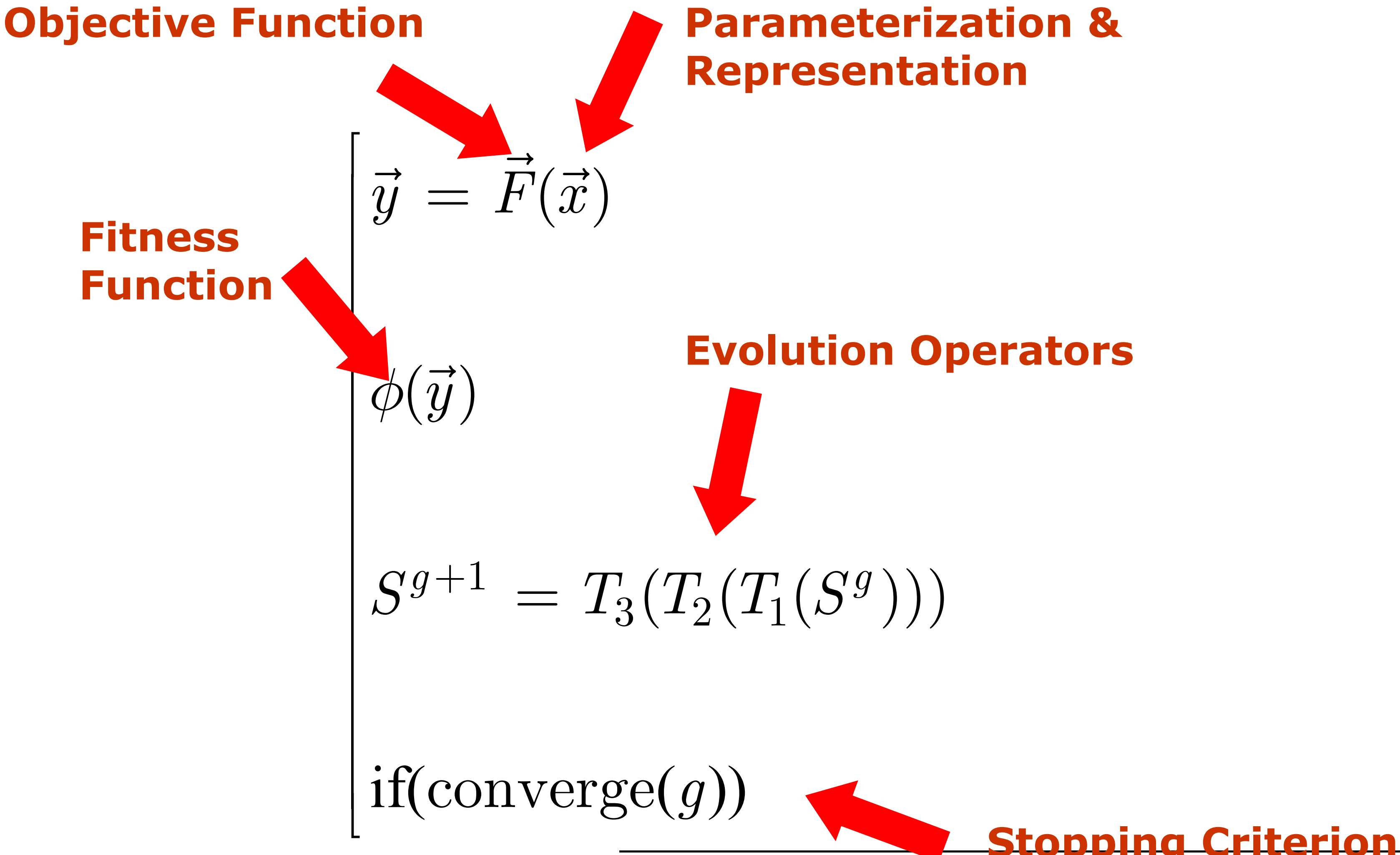
EA – Schematic Presentation:

$$g = 0, \;\; S^g = S^{random}$$



$$\left[\begin{array}{l} \vec{y} = \vec{F}(\vec{x}), \;\; \forall \; s \in \; S^g \\ \phi(\vec{y}), \; \forall \; s \; \in \; S^{g,\mu} \\ S^{g+1} = T_3(T_2(T_1(S^g))) \\ g = g + 1 \\ \text{if(converge(}g\text{)) end} \end{array}\right.$$

EA – Prerequisites:



$$\vec{y} = \vec{F}(\vec{x})$$

$$\phi(\vec{y})$$

$$S^{g+1} = T_3(T_2(T_1(S^g)))$$

$$\text{if}(\text{converge}(g))$$

Encoding the free variables – Binary Coding:

b_m, m=1,M **Binary digits per variable**

x_m **Gene**

$\vec{x}$ **Candidate solution**

Chromosome: $\underbrace{0101}_{x_1}\underbrace{101}_{x_2}\ldots..\underbrace{10111}_{x_M}$

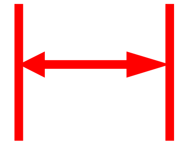
$$x_m = L_m + \frac{U_m - L_m}{2^{b_m} - 1} \sum_{i=1}^{b_m} 2^{i-1} d_{m,i}$$

L_m, U_m **user defined bounds**

Schemata in binary strings:

A <u>schema</u> is a similarity template describing a subset of strings with similarities at certain string positions (Holland, 1968)

$$*10*1$$



Defining length d(S)=5-2=3

Order of schema o(S)=3 (fixed digits)

01001	01011	11001	11011

m=length of string(=5)
r=number of *'s (=2)

2^m possible schemata

2^r strings matched by this schema

Why dealing with Schemata?

$$*10*1$$

Defining length d(S)=5-2=3

Order of schema o(S)=3 (fixed digits)

The *order* of a schema affects its survival probabilities during *mutation*

The *defining length* of a schema affects its survival probabilities during *crossover*

(Schema Theorem)

Encoding the free variable– Real Coding:

Representation: $$(x_1, x_2, \ldots, x_m, \ldots, x_M)$$

<u>A step ahead:</u>

Representation including evolution parameters (the concept of ES):

$$(x_1, x_2, \ldots, x_m, \ldots, x_M, \sigma_1, \sigma_2, \ldots, \sigma_m, \ldots, \sigma_\mathrm{M})$$

Schema Theorem (1/4) :

The effect of selection:

(m_g) examples of a particular schema (S), generation (g)

F(S)=average fitness of the strings matching schema (S)

$$m_{g+1} = m_g \frac{F(S)}{F_{mean,g}}$$

Reproductive Schema Growth Equation

$$m_{g+1} = m_g(1+k)$$

$m_{g+1}>m_g$	if $F(S)>F_{mean,g}$
$m_{g+1}<m_g$	if $F(S)<F_{mean,g}$

$$m_{g+1} = m_0(1+k)^{g+1}$$

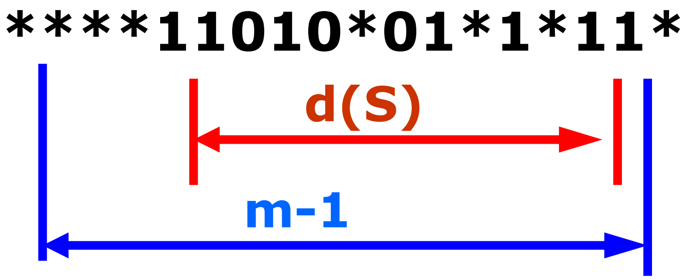
Long term effect of selection (k=constant)

Schema Theorem (2/4) :

The effect of crossover:

Possibility of destructing the schema S during crossover:

$$p_{des} = \frac{d(S)}{m-1}$$



Possibility of maintaining the schema S:

$$p_{main} = 1 - p_{Xover} \frac{d(S)}{m-1}$$

Schema Growth Equation (after selection & crossover):

$$m_{g+1}(S) \geq m_g(S) \frac{F(S)}{F_{mean,g}(S)} (1 - p_{Xover} \frac{d(S)}{m-1})$$

Schema Theorem (3/4) :

The effect of mutation:

Possibility of maintaining the schema S after mutation:

$$p_{surv} = (1 - p_{mut})^{o(S)} \approx 1 - p_{mut} o(S)$$

******11010*01*1*11***

Final Schema Growth Equation:

$$m_{g+1}(S) \geq m_g(S) \frac{F(S)}{F_{mean,g}(S)} \left(1 - p_{Xover} \frac{d(S)}{m-1} - o(S) p_{mut}\right)$$

Short, low-order, above-average schemata should receive an (exponentially) increasing number of strings in the next generations

Schema Theorem (4/4) :

Lessons Learned:

- **Short, low-order, above-average schemata sould receive an (exponentially) increasing number of strings in the next generations (Schema Theorem).**
- **GA explore the search space by short, low-order schemata.**
- **GAs seek near-optimal performance through the juxtaposition of short, low-order, high-performance schemata (the so-called building blocks, Building Block Hypothesis).**

Exploration vs. Exploitation :

- **Exploration: seeking the global optimum in new and unknown areas in the search space.**
- **Exploitation: making use the knowledge gained from the previously examined points to guide the search towards new better points in the search space.**

Holland 1975: GAs ✓

but:

- **Infinite population size**
- **Fitness function value accurately reflects the utility of a solution**
- **Genes in a chromosome do not interact significantly**

EAs: Binary or Real Coding?

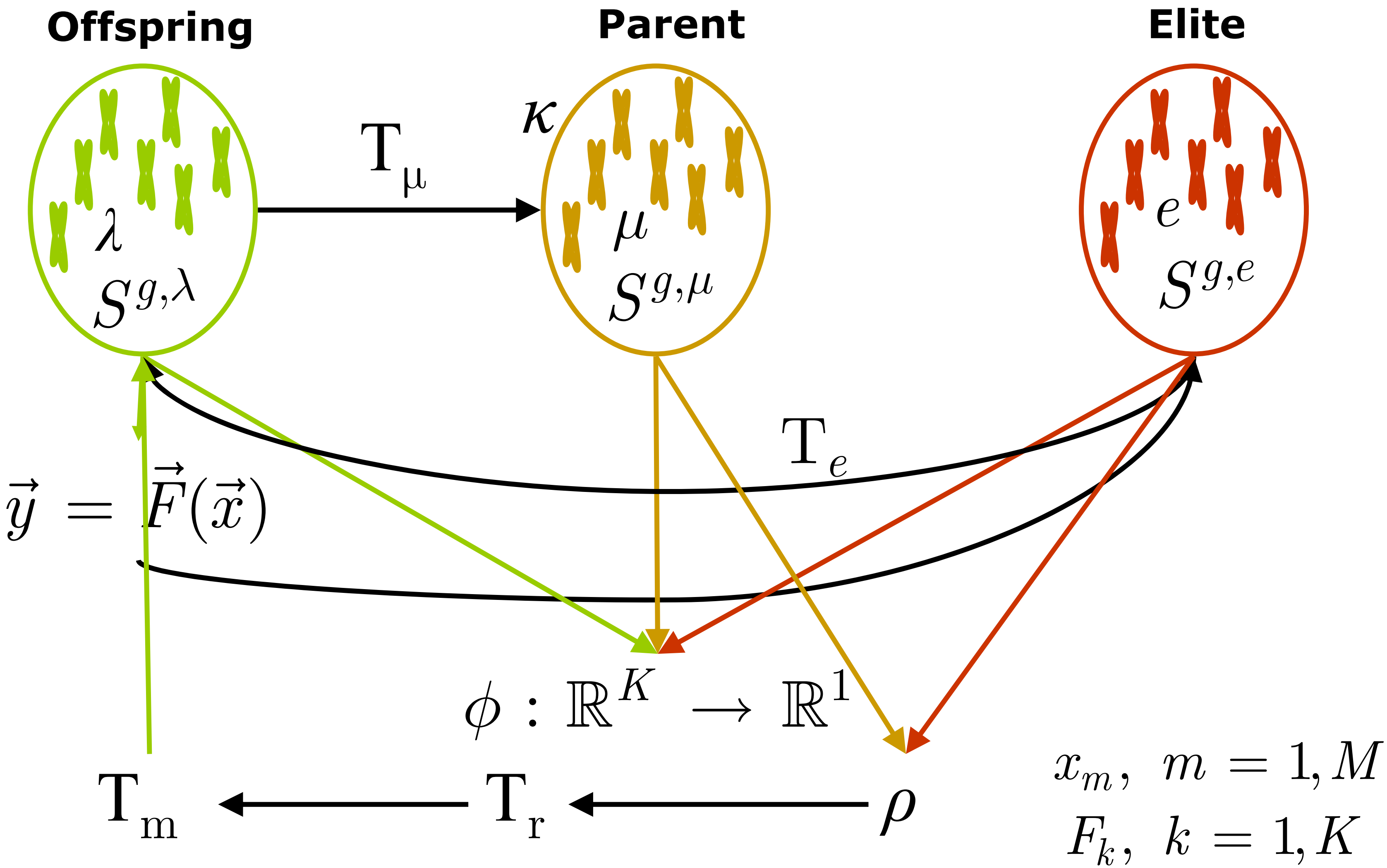
Binary Coding:

- **Creates lengthy binary strings if high accuracy is required.**
- **Offers the maximum number of schemata per bit of information, compared to any other coding.**
- **Facilitates theoretical analysis and the development of new genetic operators.**
- **Smallest alphabet that allows a natural expression of the problem (Goldberg, 1989).**

Real Coding:

- **Problem-tailored genetic operators can readily be devised.**
- **High-cardinality alphabets contain more schemata (Antonisse, 1989).**

Generalized Evolutionary Algorithm (EA)



EA – Schematic Presentation:

$$g=0,\ S^{g,e}=\varnothing,\ S^{g,\mu}=\varnothing,\ S^{g,\lambda}=S^{random}$$

$$\left[\begin{array}{l}
\vec{y}=\vec{F}(\vec{x}),\ \ \forall\ s\in\ S^{g,\lambda} \\
S^{g+1,e}=T_e(S^{g,\lambda}\cup S^{g,e}) \\
\phi(\vec{y}),\ \forall\ s\ \in\ \left(S^{g,\mu}\cup S^{g,\lambda}\cup S^{g+1,e}\right) \\
S^{g,\lambda}=T_{e_2}(S^{g,\lambda}\cup S^{g+1,e}) \\
S^{g+1,\mu}=T_\mu\left(S^{g,\mu}\cup S^{g,\lambda}\right) \\
S^{g+1,\lambda}=\mathrm{T}_m\left(T_r(S^{g+1,\mu}\cup S^{g+1,e})\right) \\
g=g+1 \\
\text{if(converge(}g\text{)) end}
\end{array}\right.$$

Parent Selection Operator T_μ (1/5)

(Neglecting Elitism)

- **<u>Phase 1:</u>**

$$S^{prov,\mu} = \mathrm{T}_{\mu,1}\left(S^{g,\mu} \cup S^{g,\lambda}\right)$$

Life-span κ
Fitness φ
(ES)

- **<u>Phase 2:</u>**

$$S^{g+1,\mu} = \mathrm{T}_{\mu,2}\left(S^{prov,\mu}\right)$$

Selective Pressure (Fitness φ)
(GA)

Parent Selection Operator T_{μ} (2/5)

- **Proportional selection**
- **Linear ranking**
- **Roulette wheel**
- **Probabilistic Tournament selection**
- **Indirect Selection ($\mu<\lambda$)**

$$S^{g+1,\mu} = \mathrm{T}_{\mu,1}\left(S^{g,\mu} \cup S^{g,\lambda}\right)$$

$$S^{g+1,\mu} = \mathrm{T}_{\mu,2}\left(S^{prov,\mu}\right)$$

Parent Selection Operator T_μ (3/5)

Proportional selection:

Select (λρ) individuals out of (μ) preselected ones

$$\frac{\mu\phi^{(s)}}{\sum_{i=1}^{\mu}\phi^{(\iota)}} \geq 1$$

Premature convergence due to the presence of a "super-fit" individual.

$$number\_of\_copies^{(s)} = \text{int}\left[\frac{\lambda\rho\mu\phi^{(s)}}{\sum_{i=1}^{\mu}\phi^{(\iota)}}\right]$$

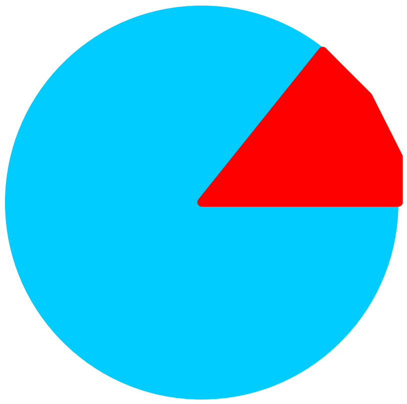
Parent Selection Operator T_μ (4/5)

Roulette Wheel:

Select (λρ) individuals out of (μ) preselected ones

Premature convergence due to the presence of a "super-fit" individual.

$$angle\_slot = \frac{\phi^{(s)}}{\sum_{i=1}^{\mu} \phi^{(\iota)}} 360^o$$



Parent Selection Operator T_μ (5/5)

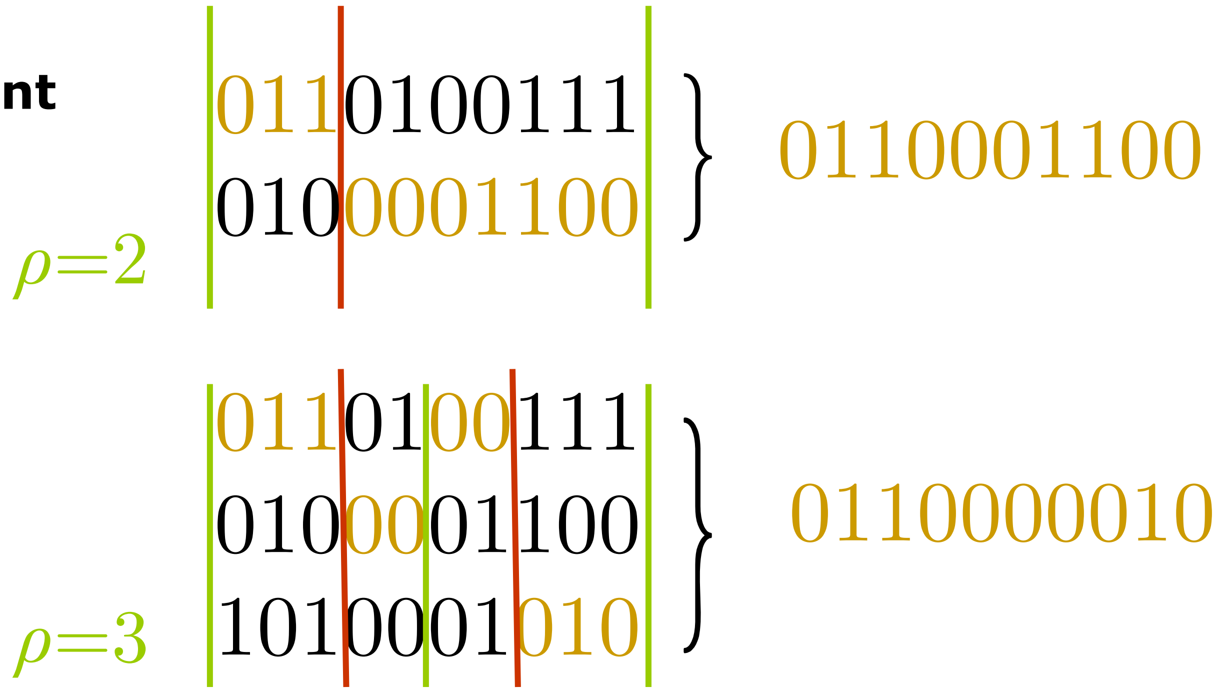
Fitness Ranking:

- **Individuals are sorted in the order of fitness values**
- **Reproductive trials are assigned according to rank**
- **Linear Ranking**
- **Exponential Ranking, etc**

- **Overcomes the problem of the presence of an extreme individual**
- **Fitness ranking performs better than fitness scaling.**

Recombination Operation T_r – Binary Coding

- **One-point**



- **Two-point**
- **One- or two-point per variable**
- **Discrete (variables' interchanging)**
- **Uniform (with parent-depending probability)**

Recombination Operation T_r – Real Coding

- **One-point**, $\rho=2$

$$\left.\begin{matrix} x_1 x_2 x_3 x_4 x_5 x_6 \\ x_1 x_2 x_3 x_4 x_5 x_6 \end{matrix}\right\} \quad x_1 x_2 x_3 x_4 x_5 x_6$$

$$x_4 = x_4 + r\,(x_4 - x_4),\ r \in [0,1]$$

- **Two-point**
- **M-point** $\quad x_m = x_m + r_m\,(x_m - x_m),\ m=1,M,\ r_m \in [0,1]$
- **Discrete**, $\rho=2$
- **Discrete Panmictic** $\boldsymbol{\rho=M}$ **(50% selection probability from each parent** $\{1,2\}, \{1,3\}, \ldots, \{1,M\}$ **)**
- **Generalized Intermediate Panmictic**

$$x_m = x_m + r_m\,(x_{m,\rho} - x_m),\ m=1,M,\ r_m \in [0,1]$$

- **Blend Xover (BLX-a),** $c=(1+2a)r-a,\ a=0.5$

Mutation Operator T_m - Binary Coding

$$P_m \sim \frac{1 \ldots 5}{\sum_{m=1}^{M} b_m}$$

0110001100 →
0010101100

<u>Dynamic Adjustment</u> of P_m, depending on

- **the number of generations without any improvement**
- **the number of evaluations without any improvement**

Mutation Operator T_m – Real Coding

Dynamic mutation probability per variable P_m

$$x_m = \begin{cases} x_m + D(g, U_m - x_m), & r_1 < 0.5 \\ x_m - D(g, x_m - L_m), & r_1 \geq 0.5 \end{cases}$$

$$D(g,a) = a \cdot r_2 \cdot \left(1 - \frac{g}{g_{\max}}\right)^p$$

ή

$$D(g,a) = a \cdot \left(1 - r_2^{\left(1-\frac{g}{g_{\max}}\right)^p}\right)$$

$p \sim 0.2$

… or, using number of evaluations, instead of number of generations

Mutation Operator T_m – Real Coding

$$\sigma_m = \sigma_m \cdot \exp\left(\tau' \cdot \mathrm{N}(0,1) + \tau \cdot \mathrm{N}_m(0,1)\right)$$

$$x_m = x_m + \sigma_m \cdot N_m(0,1)$$

$$\tau = \left(\sqrt{2\sqrt{\mathrm{M}}}\right)^{-1}$$

$$\tau' = \left(\sqrt{2\mathrm{M}}\right)^{-1}$$

Genetic Algorithms or Evolution Strategies or …

GA

- **Binary Coding**
- **$\mu=\lambda$, parents=offspring**
- **$\rho=2$, two-parent recombination**
- **$\kappa=0$, zero life-span**
- **$P_r<1$, recombination probability**
- **$K=1$, one target**

[Holland, 1970]
[Goldberg, 1989]
[Michalewicz, 1994]
[Fogel]

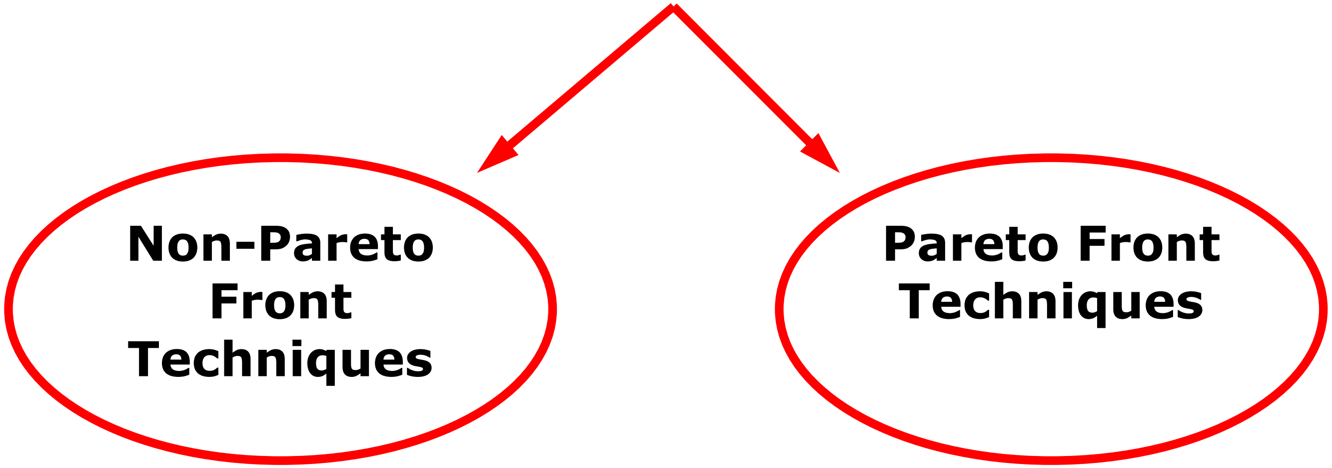
ES

- **Real Coding, including the evolution parameters**
- **Without parent Selection Operator ($\mu<\lambda$)**
- **$\mu<<\lambda$**
- **$\kappa=0$ $(\mu,0,\lambda) = (\mu,\lambda)$ or $\kappa=\infty$ $(\mu,\infty,\lambda) = (\mu+\lambda)$**
- **$\rho=2$**
- **$P_r=1$**
- **$K=1$**

[Schwefel, Rechenberg, 1965]

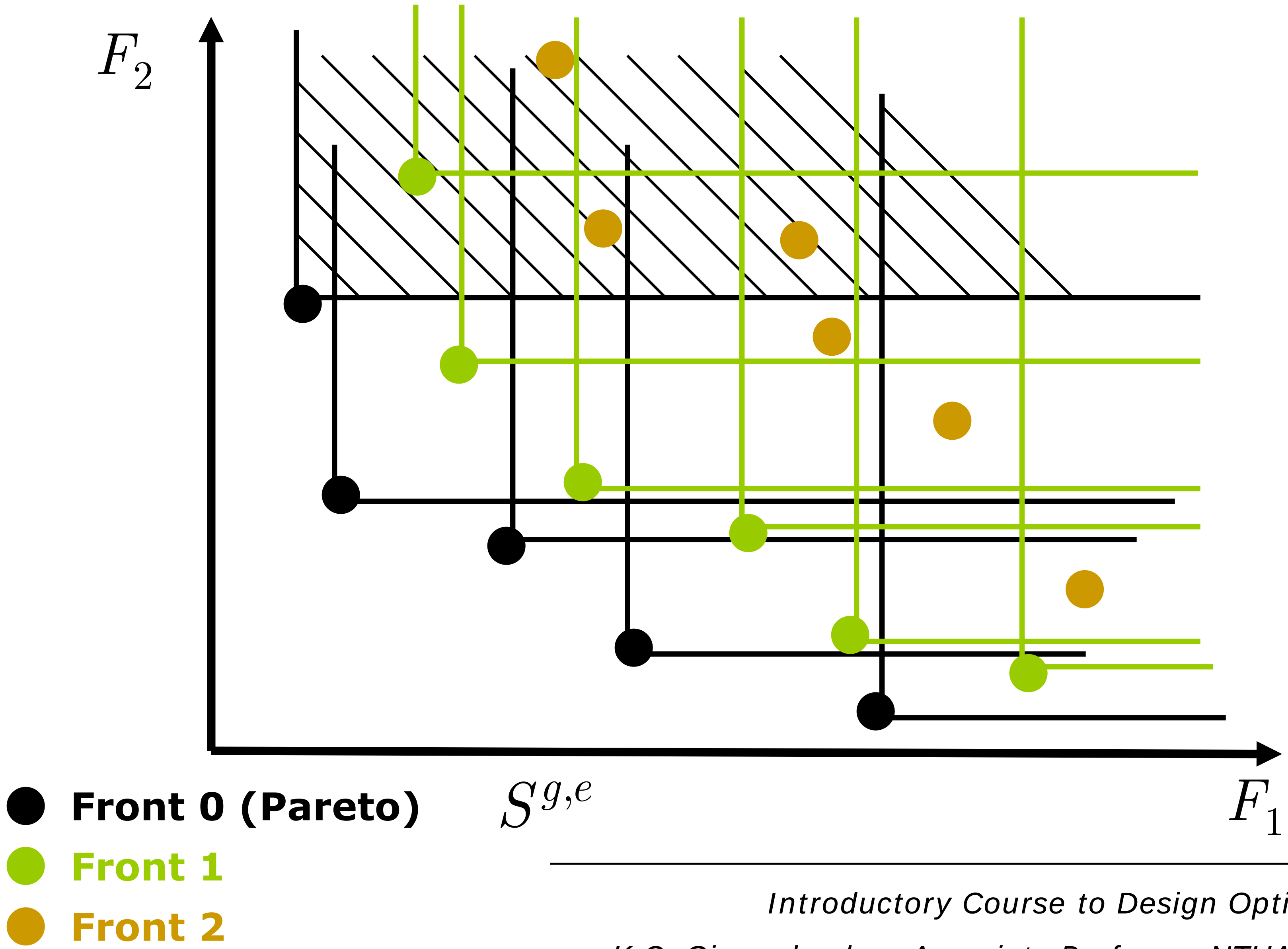
[Bäck, 1996]

Multi-objective Optimization



(K = number of objectives)

Pareto Front



Pareto Front – Definition (Minimization Problems):

Dominant Solution:

$$\vec{x}^{(p)} < \vec{x}^{(q)} \Leftrightarrow s^{(p)} < s^{(q)} \Leftrightarrow$$

$$\forall\ k \in \{1,\dots,K\} : F_k^{(p)} \le F_k^{(q)} \quad \wedge$$

$$\exists\ k \ \in \{1,\dots,K\} : F_k^{(p)} < F_k^{(q)}$$

Pareto Optimal Solution:

$$\vec{x} \Leftrightarrow \nexists\ \vec{x}' \in \mathbb{R}^M : \vec{x}' > \vec{x}$$

Multi-Objective Optimization – Computation of φ

$$\phi = \sum_{k=1}^{K} w_k \cdot F_k$$

$$\phi = F_k,\ \ k \in [1, K]$$

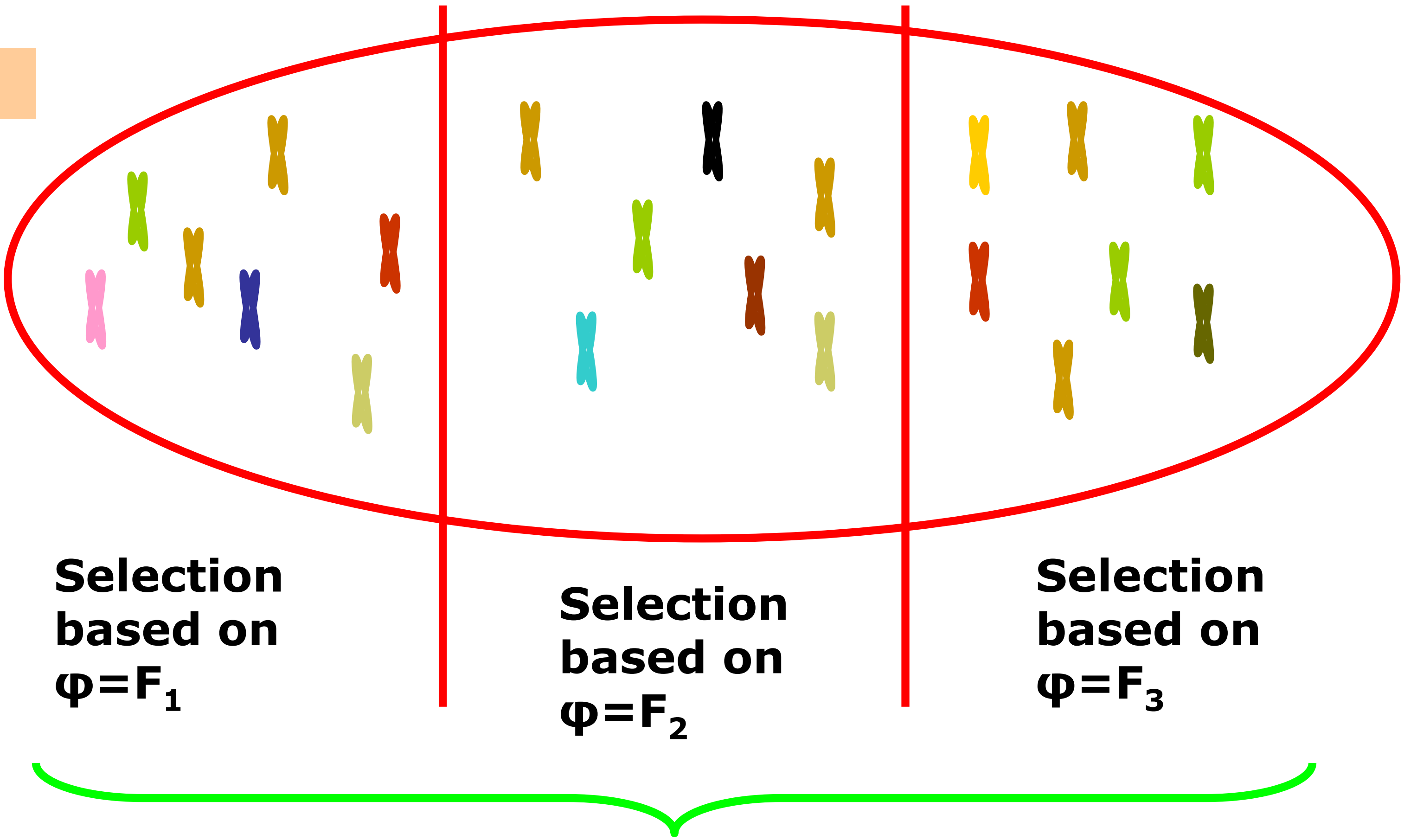
For parent selection, VEGA [Schaffer, 1984]

$$\phi = \text{Pareto\_Rank}(F_k),\ \ \text{k} = 1, K$$

[Goldberg, 1989]

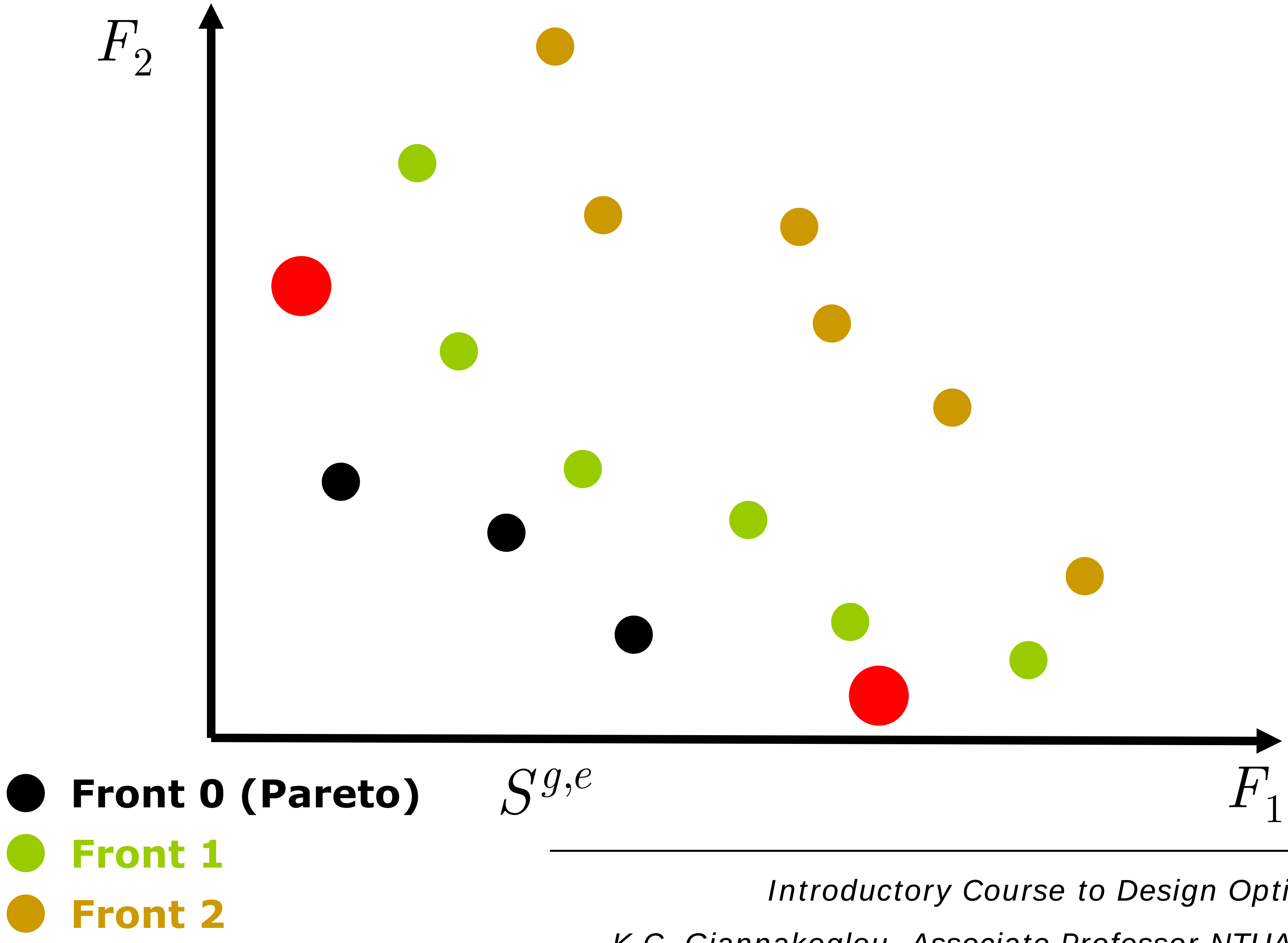
Vector Evaluated Genetic Algorithm (VEGA)

K=3



Recombination, Mutation

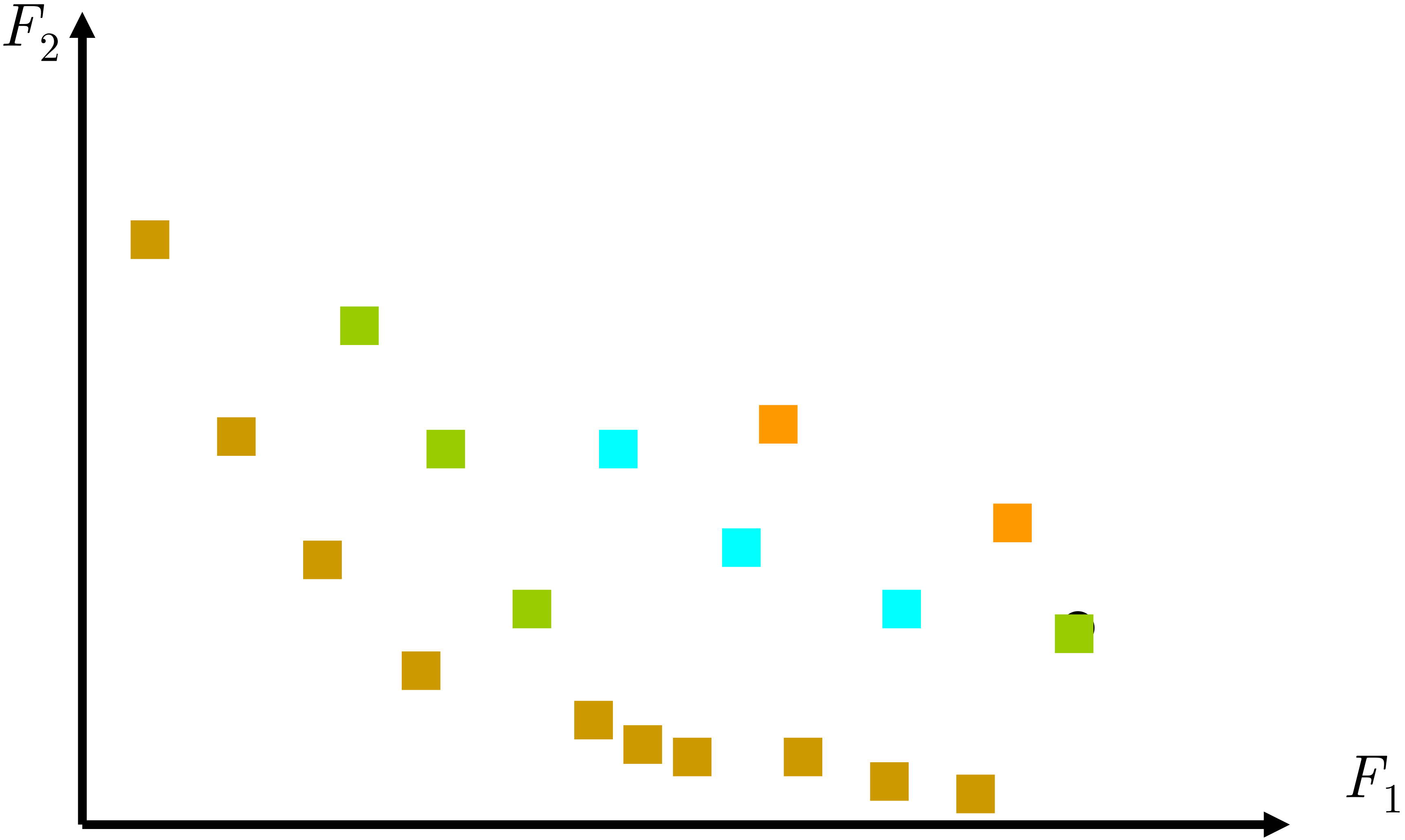
VEGA: Guess the final solutions…



φ computation using the Pareto front:

- **Front Ranking [Goldberg, 1989]**
- **Niched Pareto GA (NPGA) [Horn, Nafpliotis, 1993]**
- **Nondominated Sorting GA (NSGA) [Srinivas, Deb, 1994]**
- **Strength Pareto EA (SPEA) [Zitzler, Thiele, 1998]**
- **Pareto Envelope-based Selection Algorithm (PESA) [Corne, Knowles, Oates, 2000]**
- **NSGA-II [Deb, Agrawal, Pratap, Meyarivan, 2000]**
- **Strength Pareto EA II (SPEA II) [Zitzler, Laumanns, Thiele, 2001]**

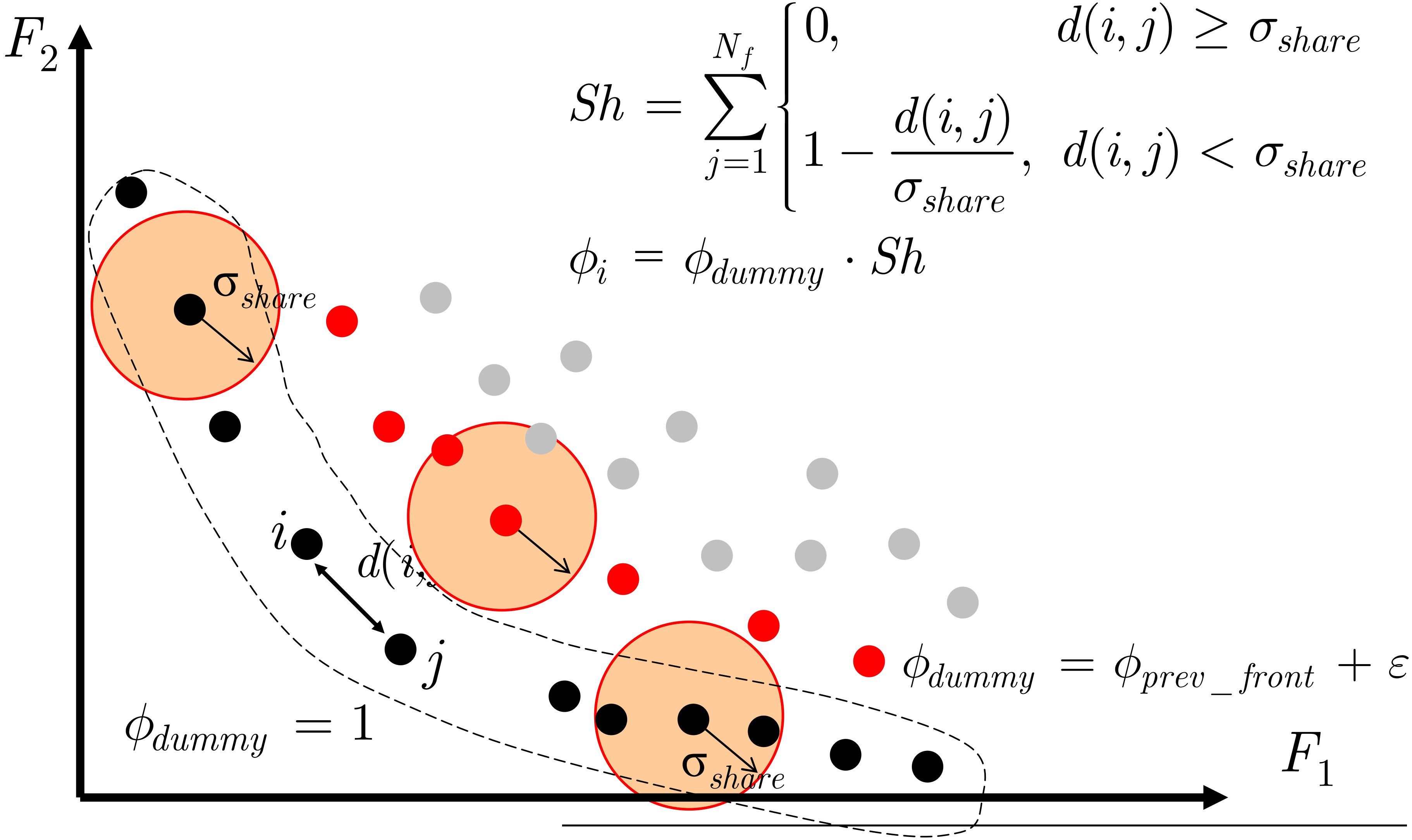
φ computation – Sorting (NSGA)



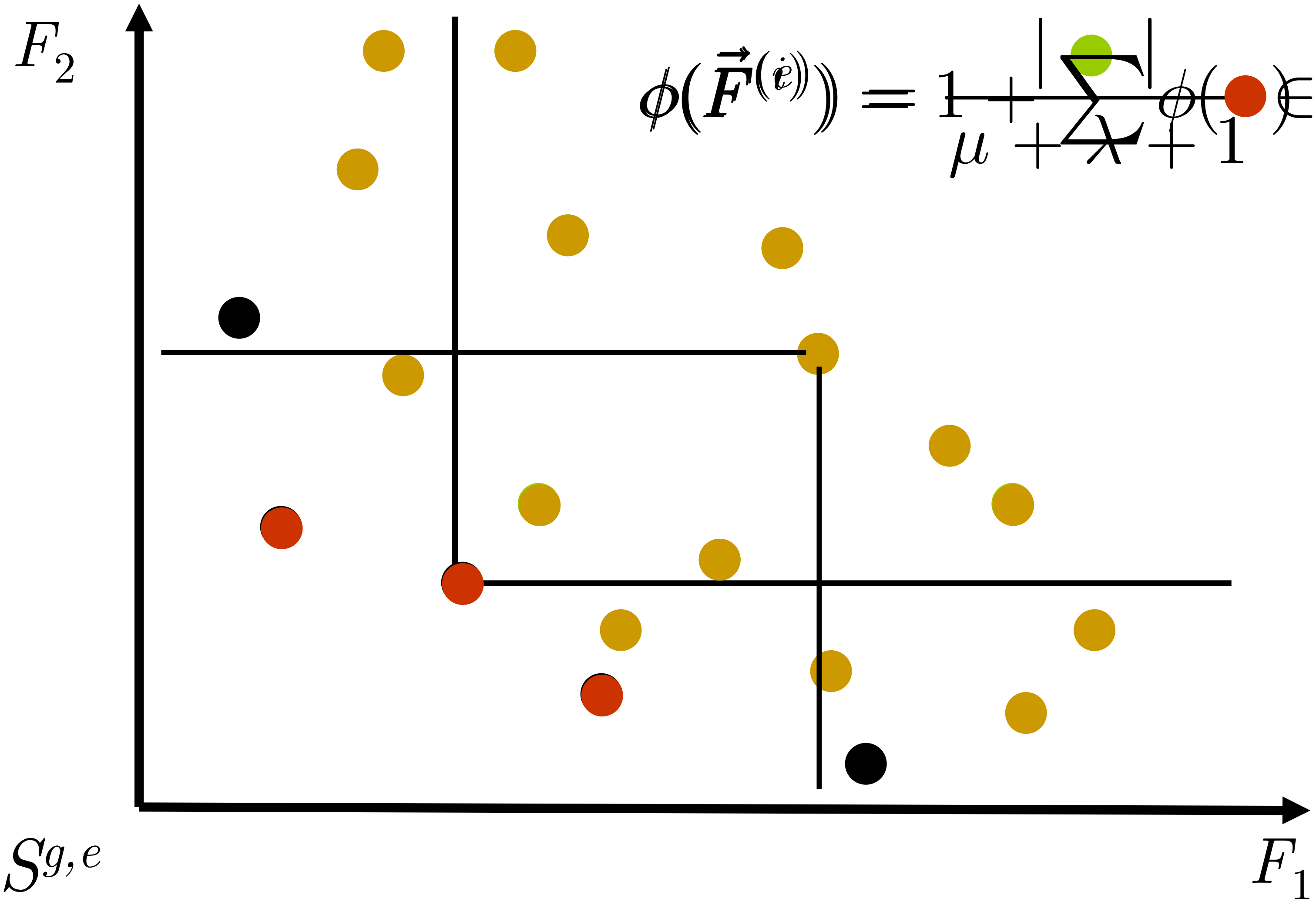
φ computation – Niching (NSGA)

$$Sh = \sum_{j=1}^{N_f} \begin{cases} 0, & d(i,j) \geq \sigma_{share} \\ 1 - \dfrac{d(i,j)}{\sigma_{share}}, & d(i,j) < \sigma_{share} \end{cases}$$

$$\phi_i = \phi_{dummy} \cdot Sh$$



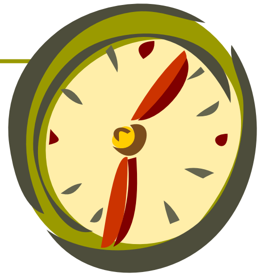
φ computation – SPEA



$$\phi(\vec{F}^{(i)}) = \frac{1+\sum \phi(\bullet)}{\mu+\lambda+1} \in [0,1)$$

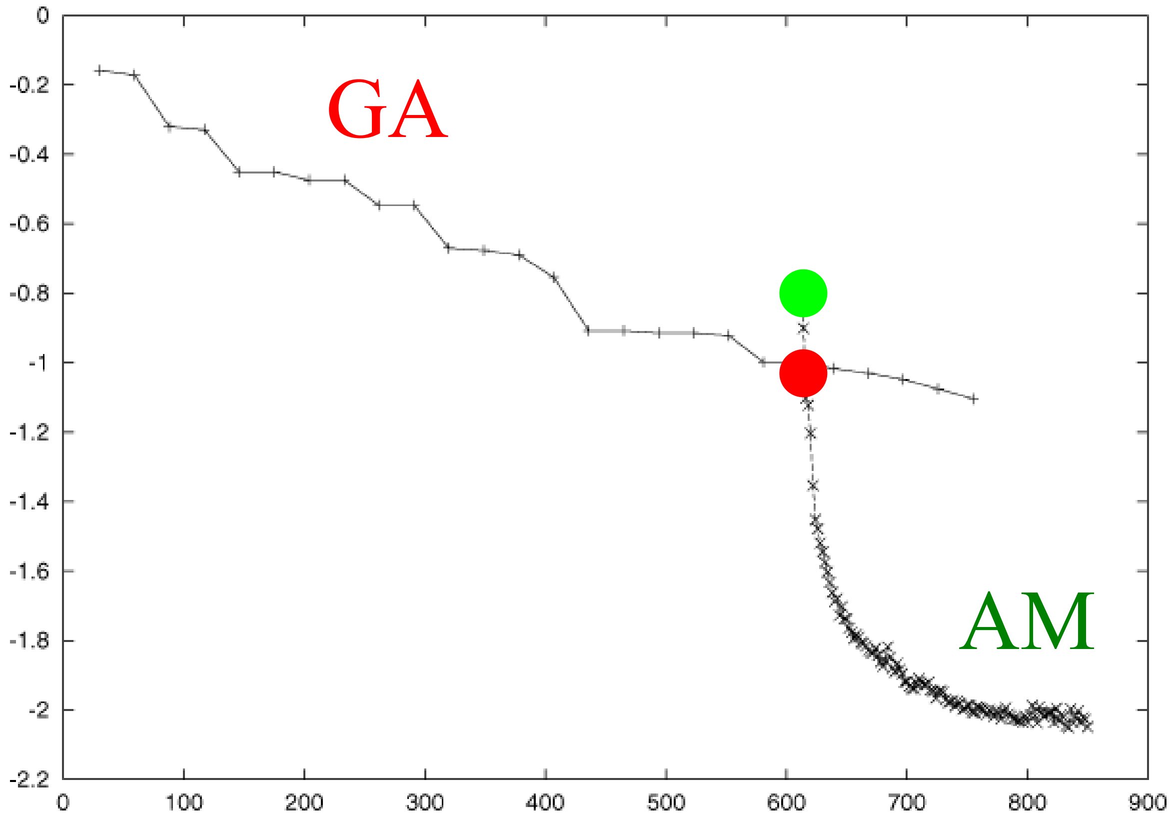
- $S^{g,e}$
- Dominated Solutions
- Other Solutions

Ways to reduce the number of evaluations:



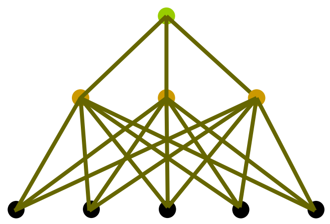
- **Improved evolution operators**
- **Hybridization with other optimization methods**
- **Distributed EAs (island model)**
- **Hierarchical EAs (faster solvers)**
- **Use of <u>surrogate models</u> (<u>metamodels</u>, fast approximate model)**

Hybridization with other Optimization Methods



Use of Surrogate Evaluation Models

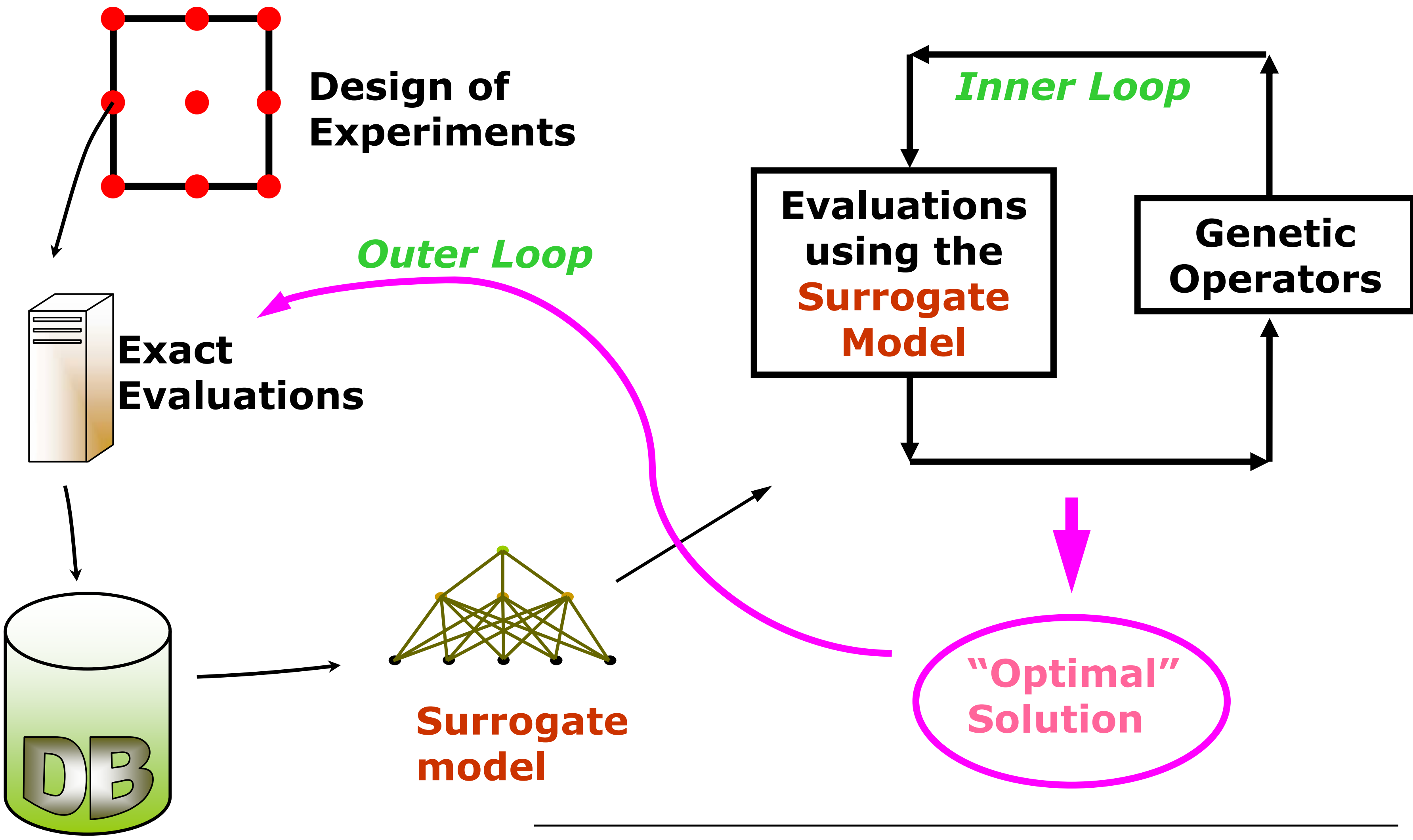
- **Polynomial Interpolation**
- **Artificial Neural Networks**
  - **Multilayer Perceptron**
  - **Radial Basis Function Networks**
- **Kriging**



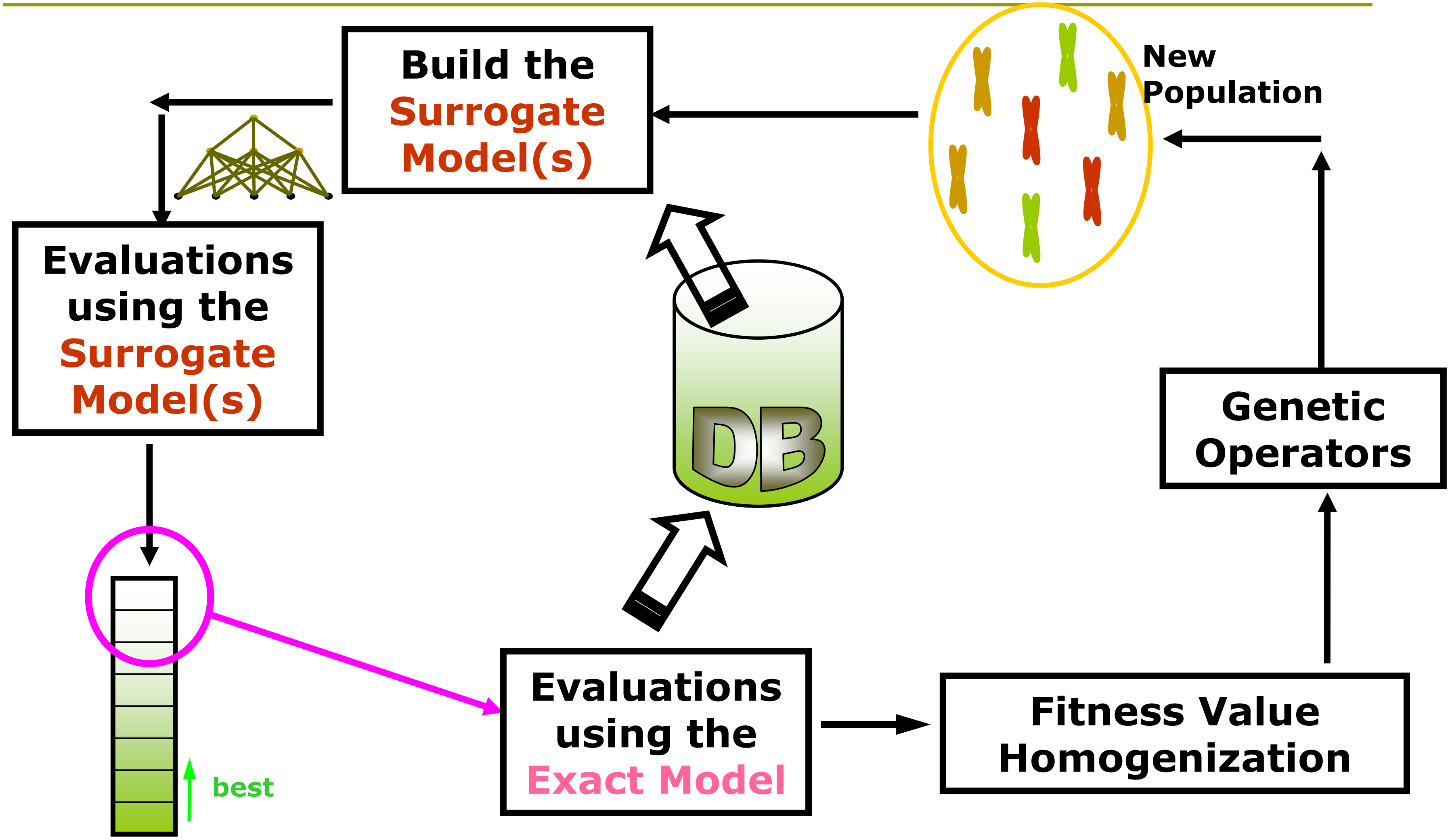
Ways of using the surrogate evaluation model:

- **Decoupled from the exact evaluation tool (+Design of Experiments, DoE)**
- **In combination with the exact evaluation tool**
  - **Regular Training (depending on the number of new entries in the DB)**
  - **Dynamical Training (separately, for each new individual)**

Use of Surrogate Models (with Off-Line Training)

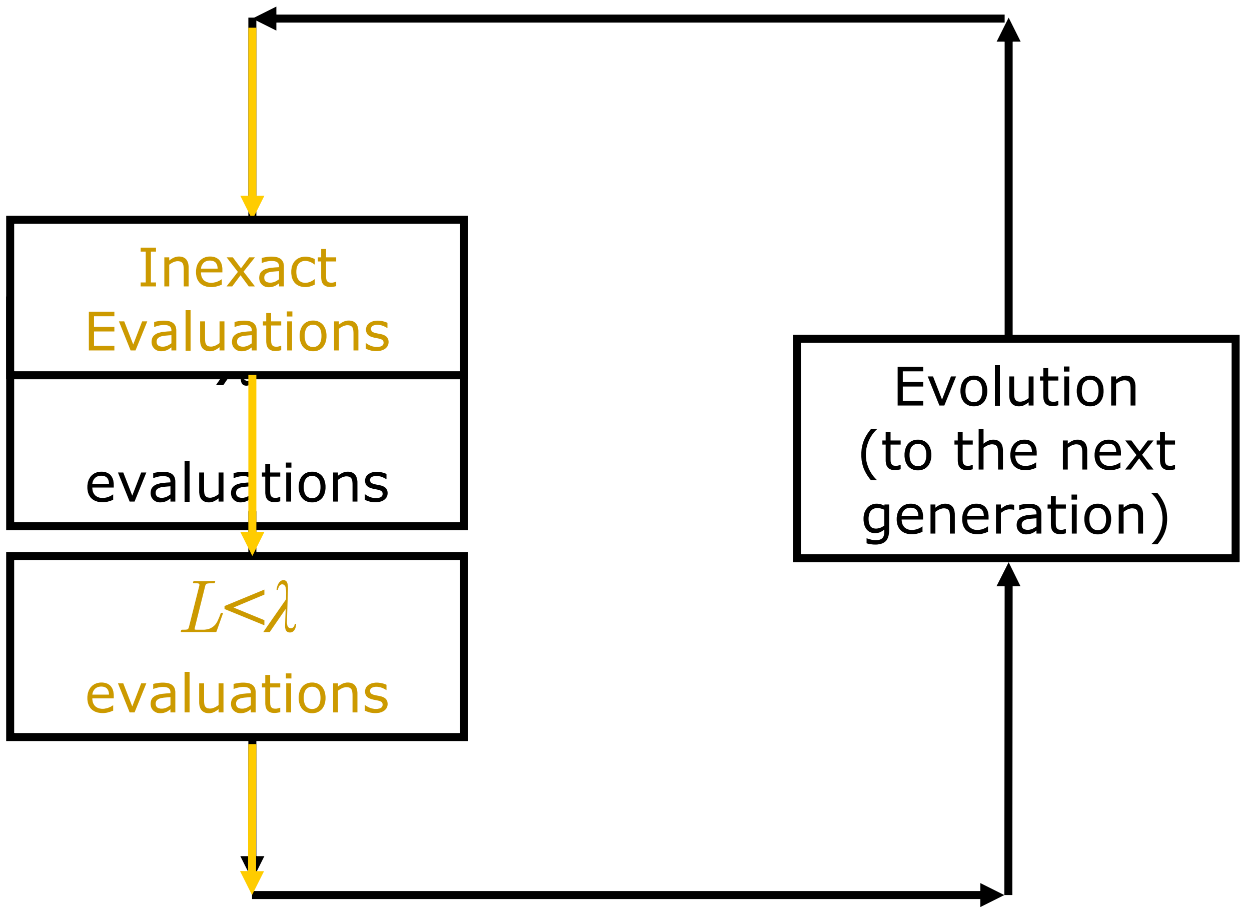


Use of Surrogate Models (with On-Line Training)



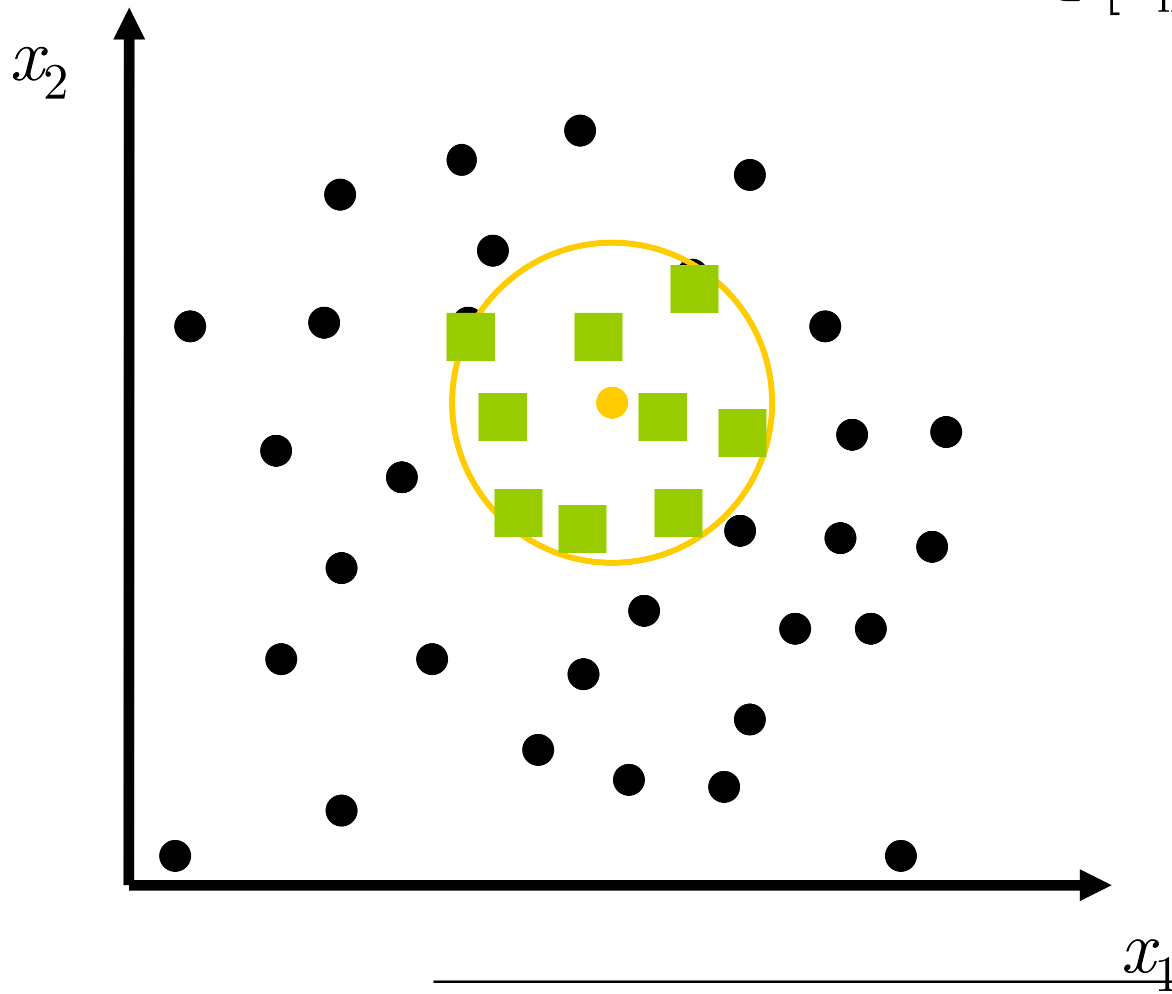
Inexact Pre-Evaluation (IPE) – The Concept:

Exact Evaluation of the most promising solutions



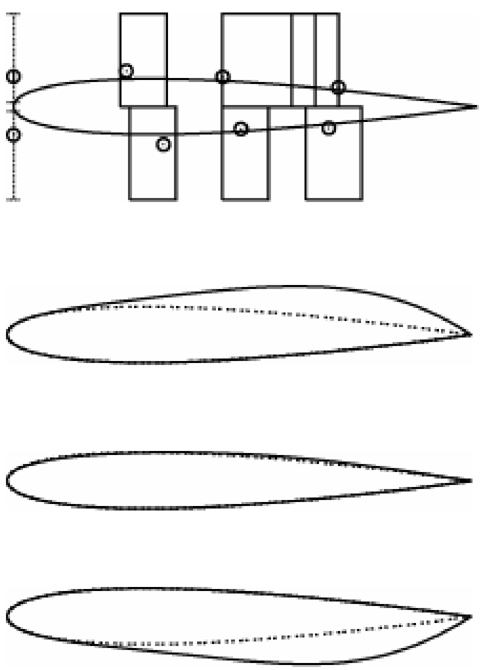
Local Surrogate Models - Training Set:

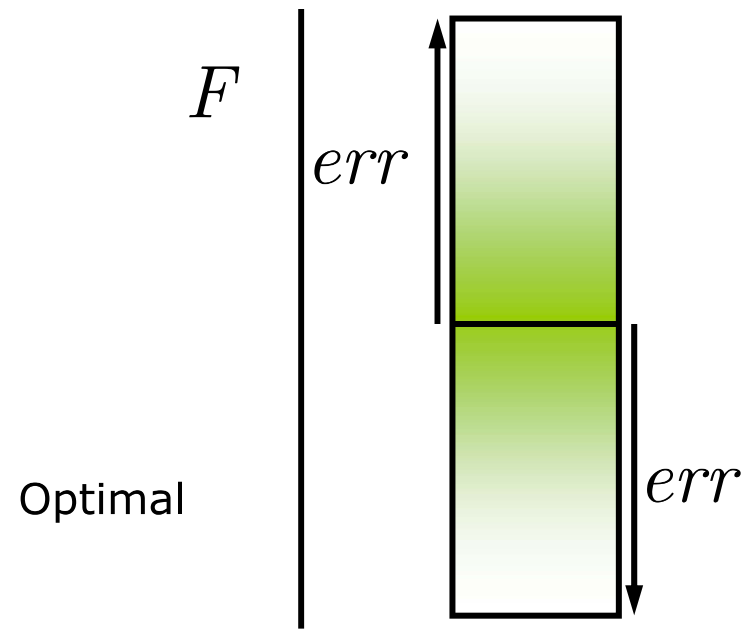
$$T \in [T_{\min}, T_{\max}]$$



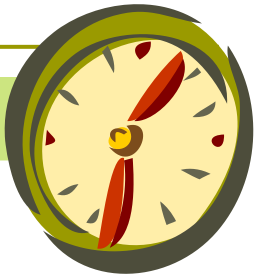
Surrogate Models – What else do they tell us???

- **Fitness Function Approximation**
- **Confidence Intervals**
- **Hessian Matrix Approximation**
- **Sensitivity Derivatives (Importance Factors)**





Optimization and Multi-processing



Reduction of the Optimization Wall-Clock Time

Parallelization Level

- **EA**
- **CFD**

EA:

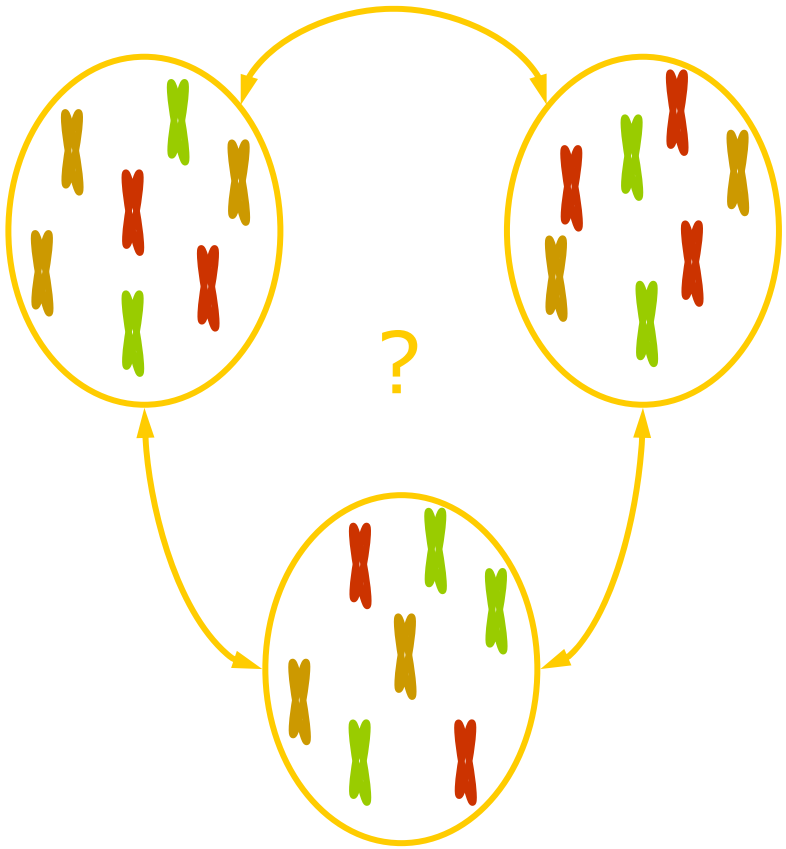
- CPUs $\neq \lambda$ (ή L)
- Loading Distribution:
  - Heterogeneous Platforms
  - Loading per processor
  - Variable evaluation cost
- Synchronization (in each generation)

- **Master: EA Module - waiting list for evaluations**
- **Slave: discrete (remote) evaluation process**

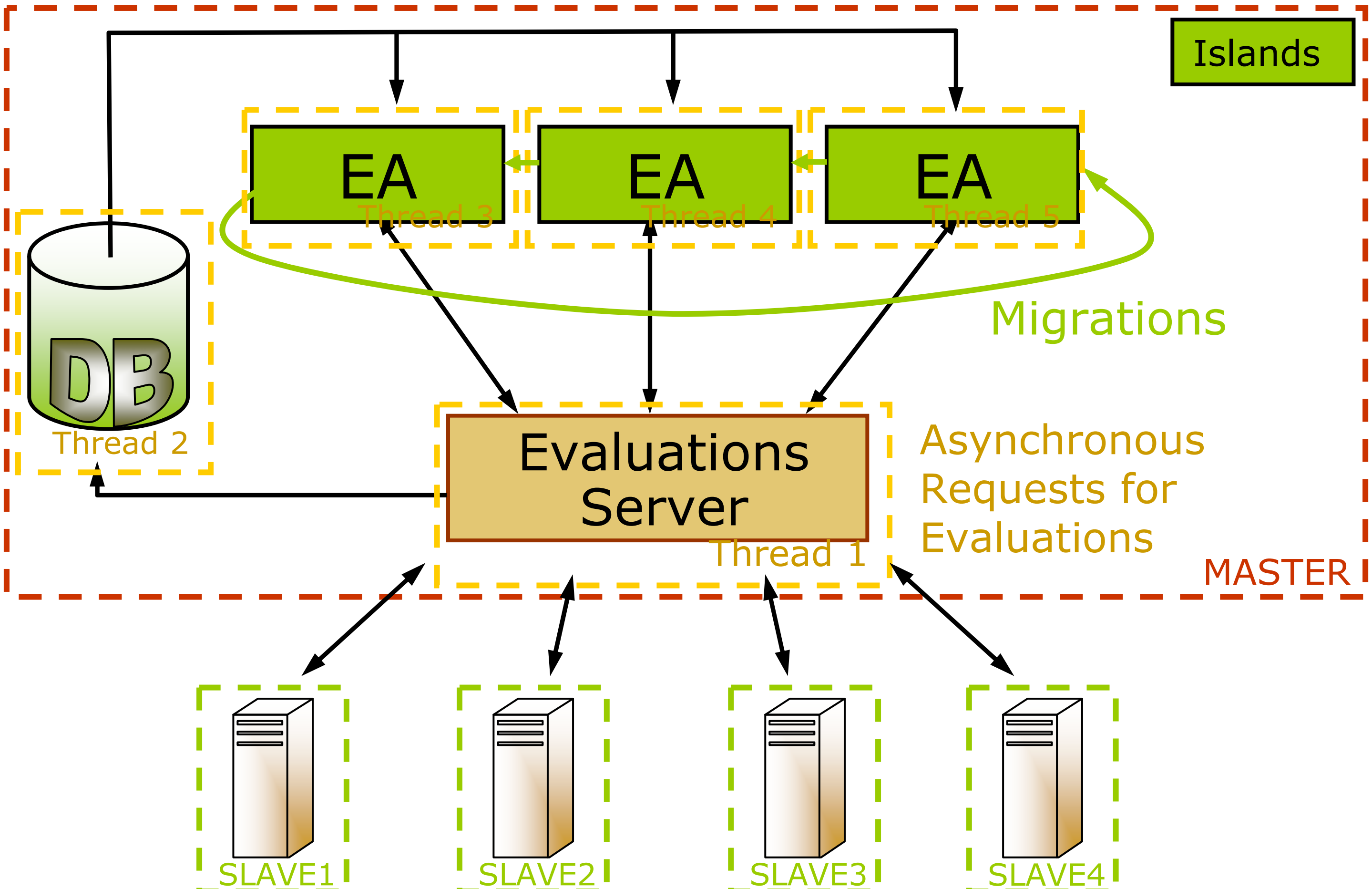
Distributed EA

- **Why ?**
  - ***L*** **(<< *λ*) < CPUs**
  - **Persistant diversity in populations**
  - **Straightforward parallelization**

- **Additional Parameters:**
  - **Number of islands**
  - **Communication topology**
  - **Communication frequency**
  - **Migration algorithm**
  - **EA parameters per island**



Distributed EA on a Multi-Processor System



Applications

EASY v1.3

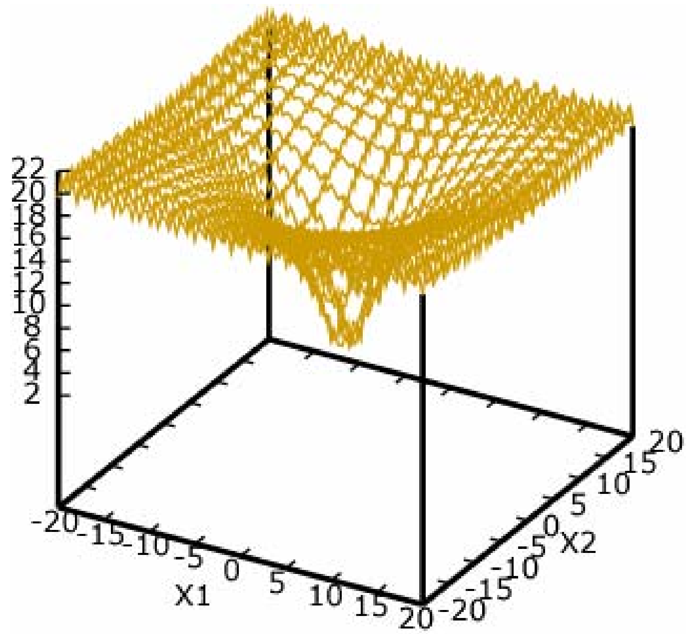
The Evolutionary
Algorithms SYstem

Developed by the National Technical University of Athens, (NTUA), Greece

Problem: Rastrigin's Function

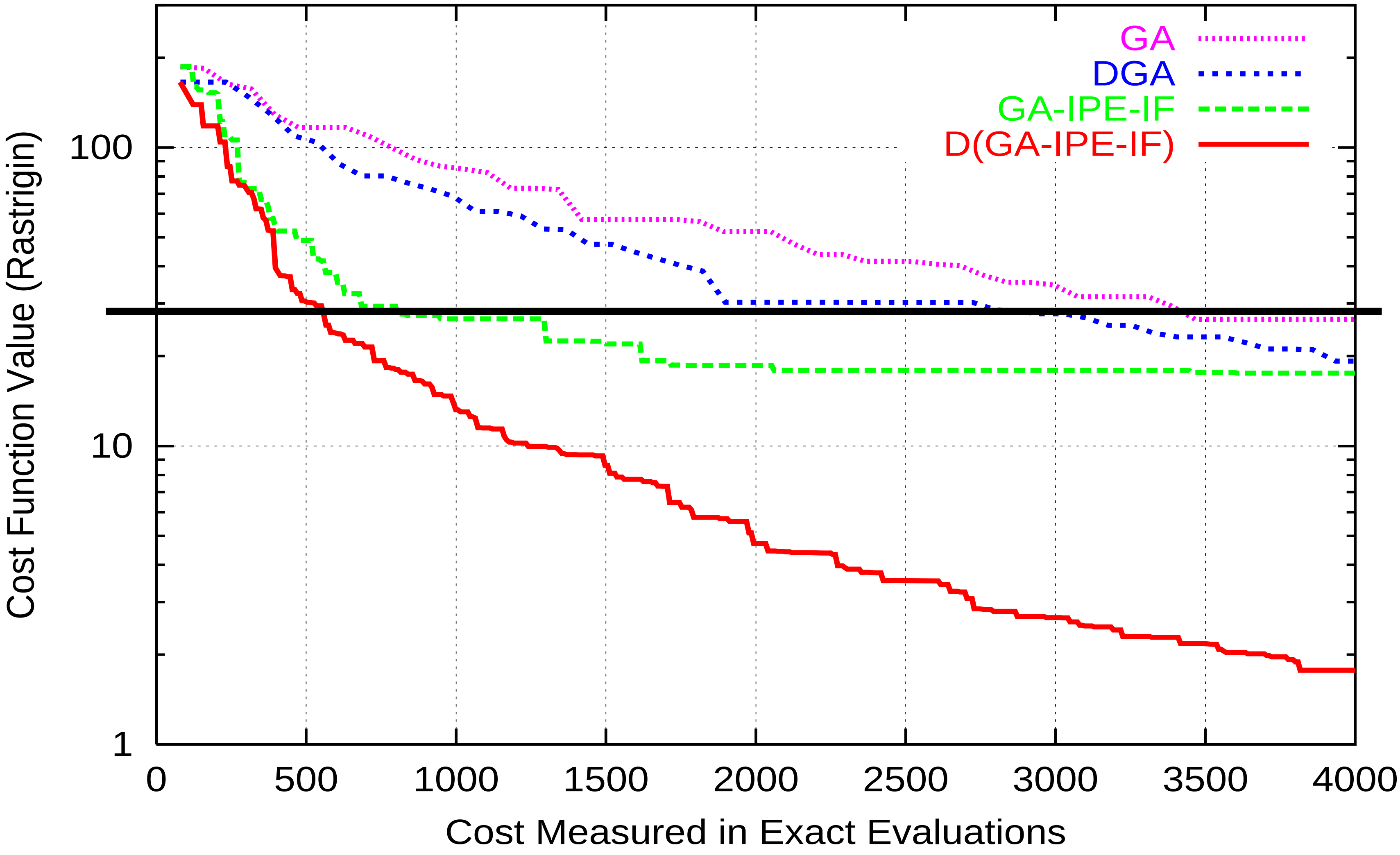
$$F(\vec{x}) = 20 + e - 20\exp(-0.2\sqrt{\frac{1}{M}\sum_{m=1}^{M} x_m^2})$$

$$-\exp(\frac{1}{M}\sum_{m=1}^{M}\cos(2\pi x_m))$$

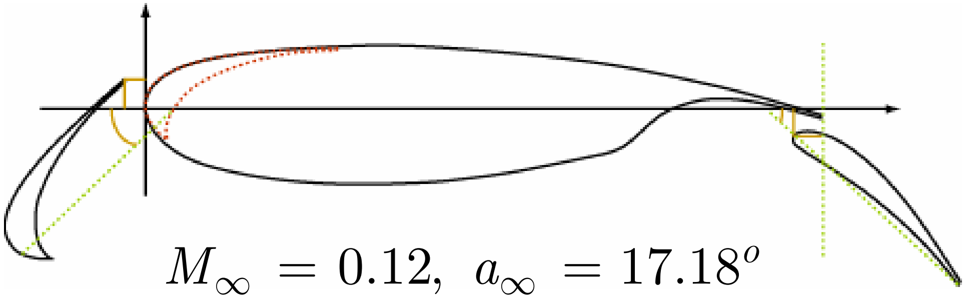


$M=30$ $\qquad -30 \leq x_m \leq 30$

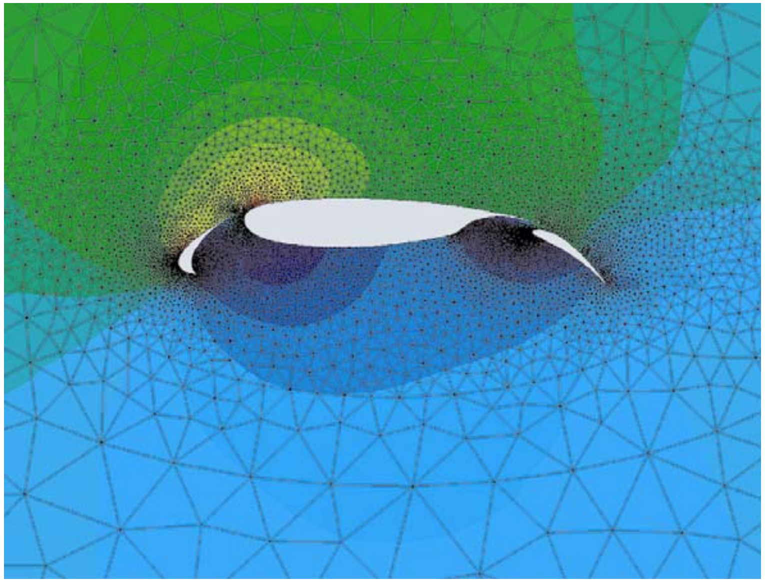
Results: Rastrigin's Function



Problem: Three-Element Airfoil, Lift Maximization

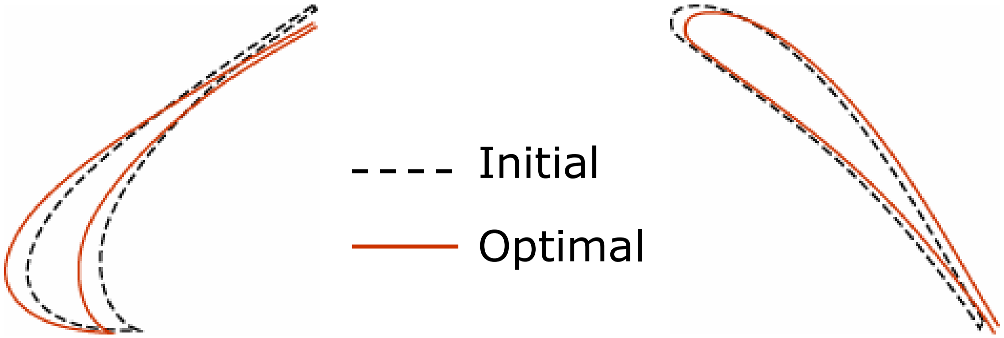


$$M_\infty = 0.12,\ a_\infty = 17.18^o$$



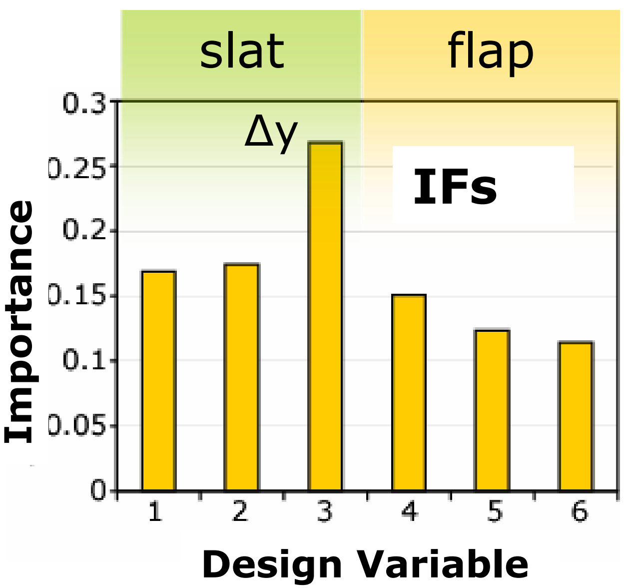
$$F = -C_L$$

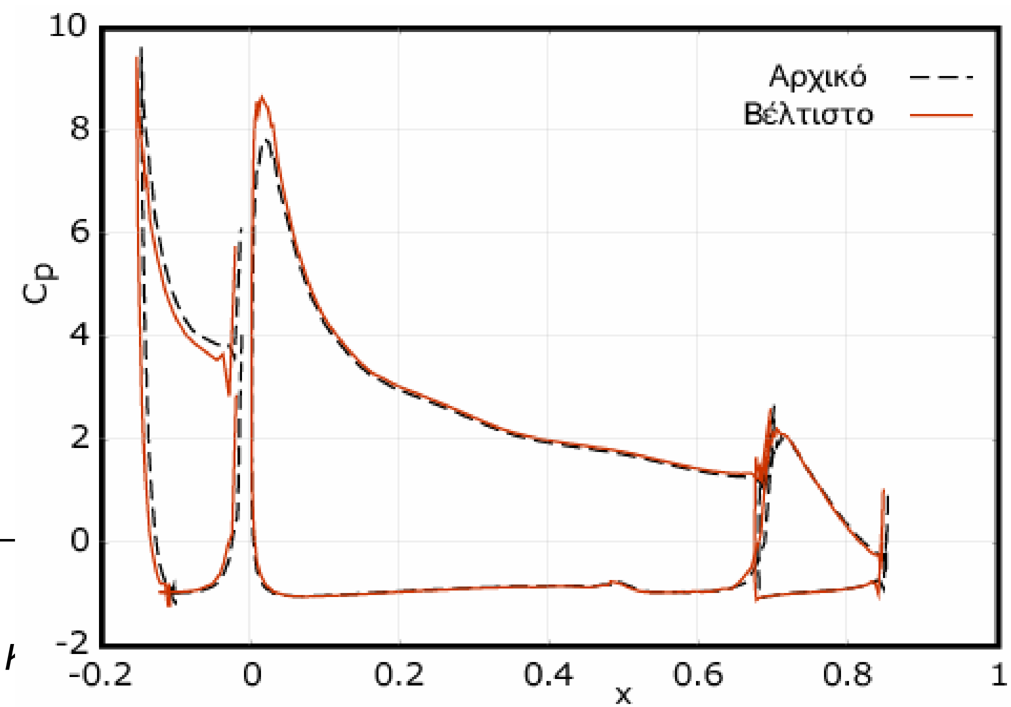
	Initial
Rotation Angle	28.1°
Δx	-0.020
Δy	0.0269
Rotation Angle	-37°
Δx	0.020
Δy	0.0249



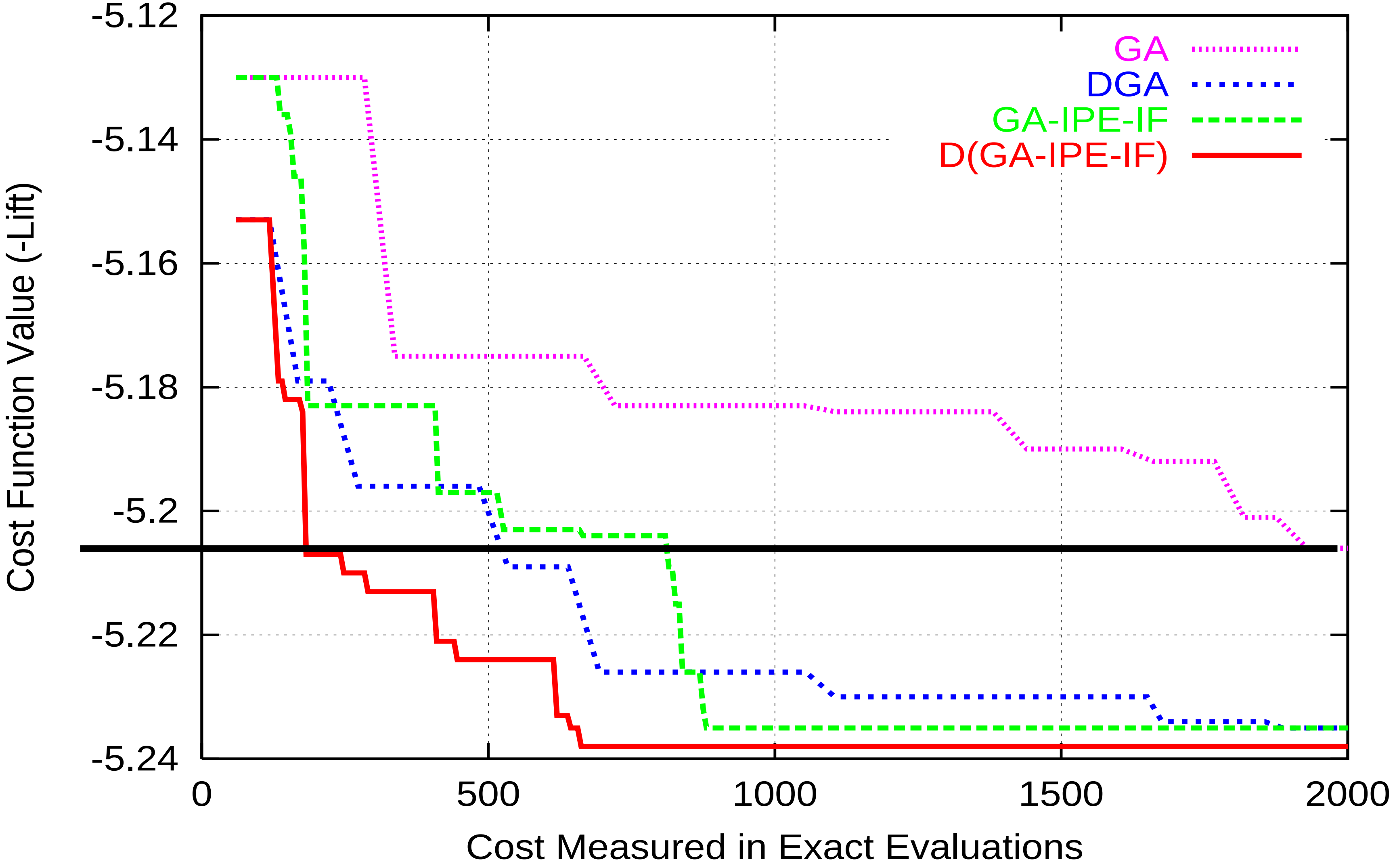
Results: Three-Element Airfoil, Lift Maximization

	Initial	Optimal
Rotation Angle	28.1°	28.02°
Δx	-0.020	-0.03078
Δy	0.0269	0.01982
Rotation Angle	-37°	-36.96°
Δx	0.020	0.02016
Δy	0.0249	0.02469



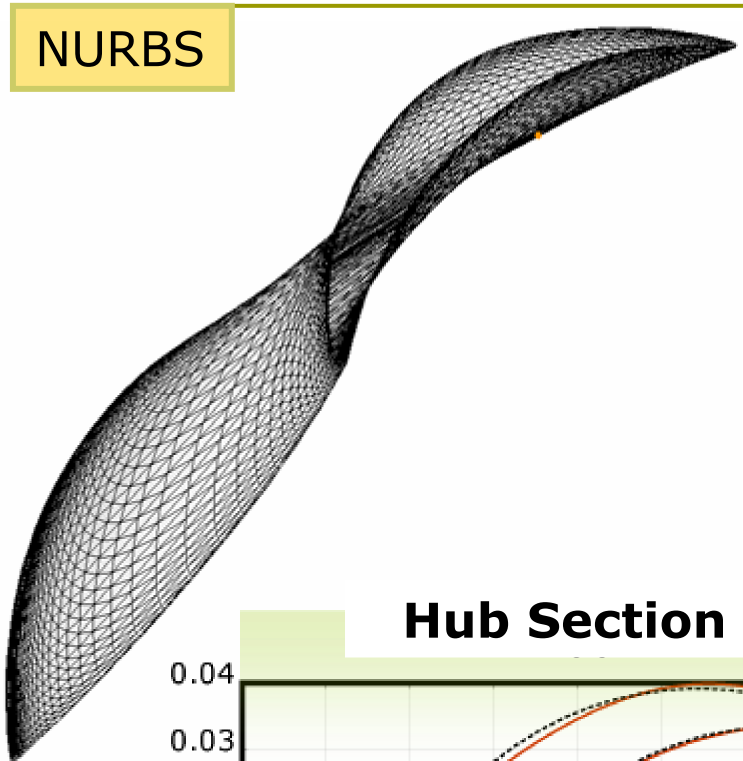


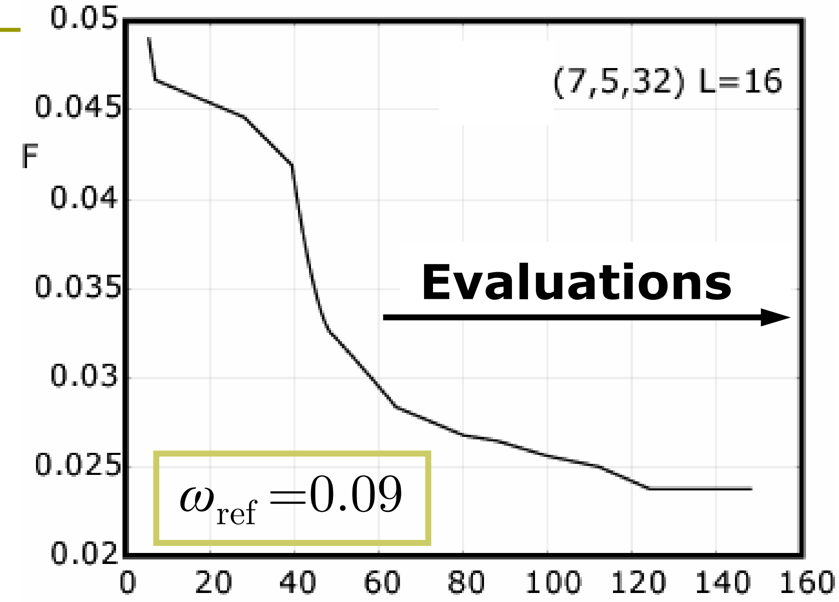
Results: Three-Element Airfoil, Lift Maximization

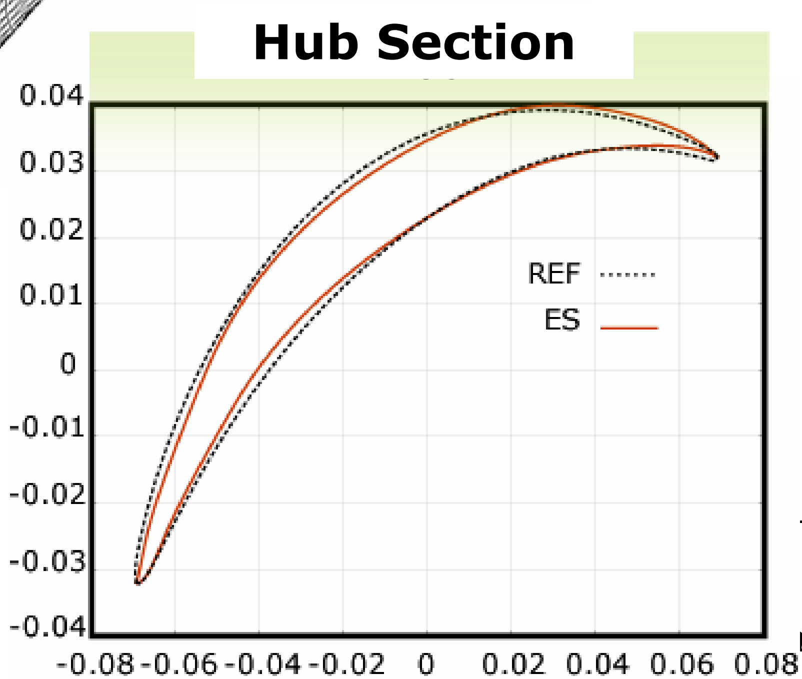


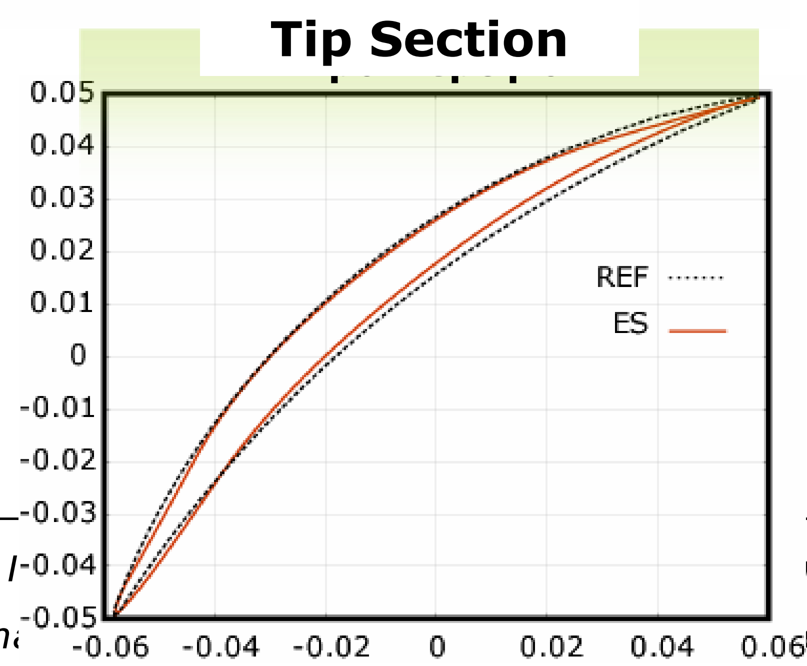
Problem: 3D Compressor Blade Design

NURBS

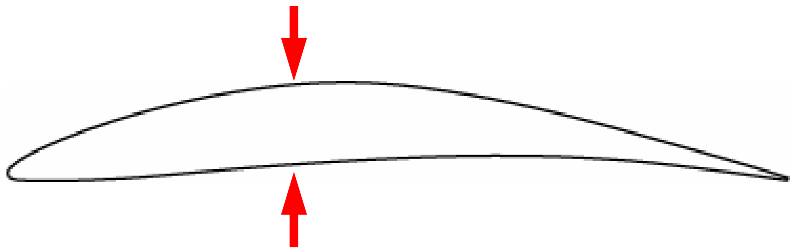








Problem: Design of a Compressor Cascade



s=0.68c γ=30º c=0.07m

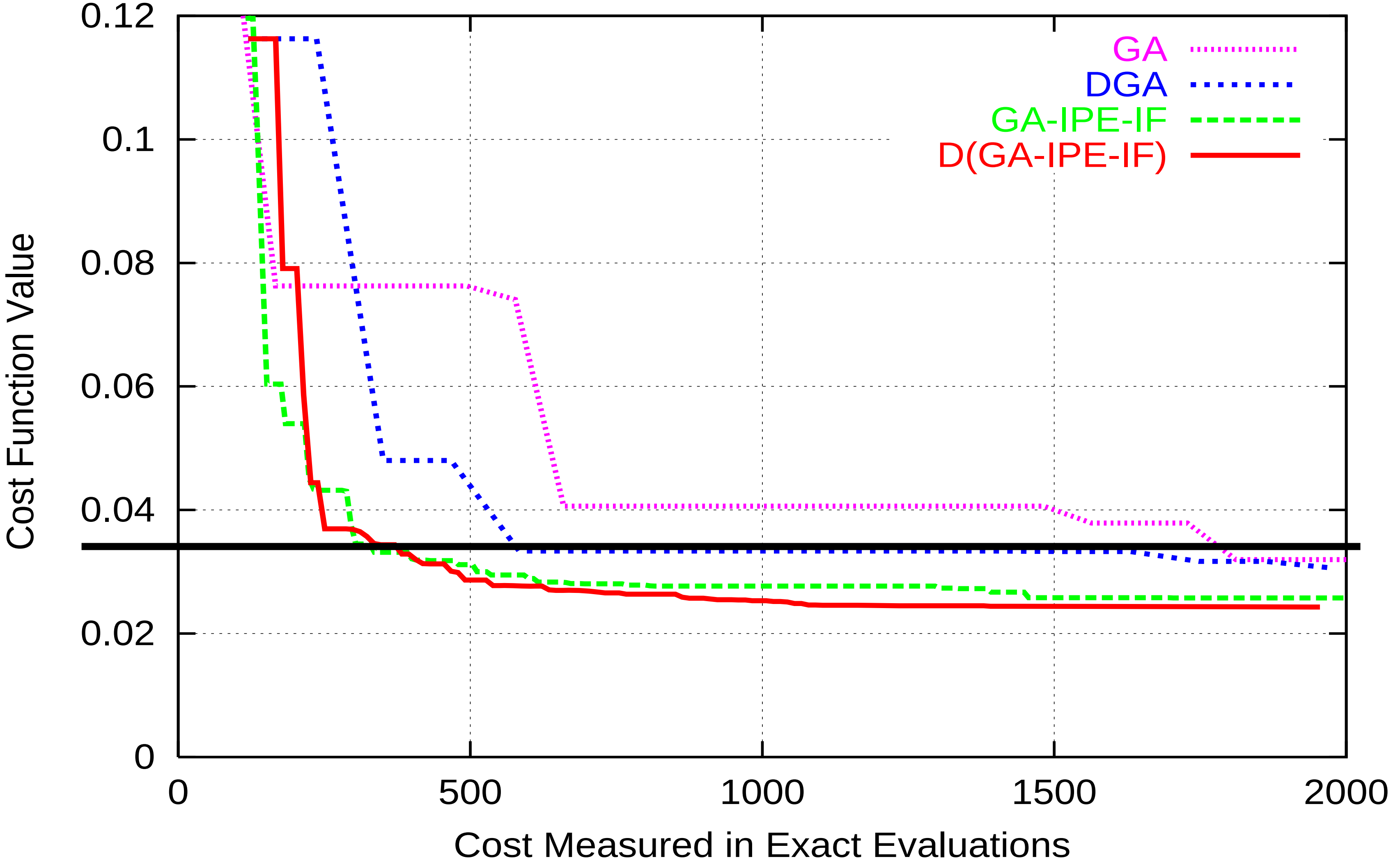
- **Minimum Total Pressure Losses**
- **Constraint on the minimum (maximum thickness)**
- **Desirable Flow turning**

$$F_1 = \omega \cdot P_1 \cdot P_2$$

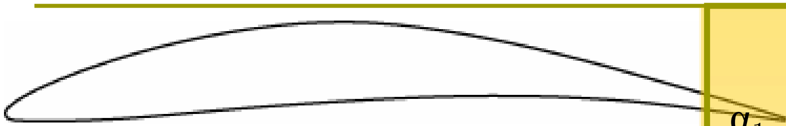
$$P_1 = \exp\left(\frac{|t_{\max} - t_{thres}|}{t_{thres}}\right), \; t_{thres} = 0.9 t_{\max,ref}, \; t_{\max} < t_{thres}$$

$$P_2 = \exp\left(-\max\left(1, \frac{\Delta\alpha_{ref} - \Delta\alpha}{\Delta\alpha_{ref}}\right)\right), \; \Delta\alpha = \alpha_1 - \alpha_2 > 0$$

Results: Design of a Compressor Cascade



Problem: Compressor Multi-Point Design

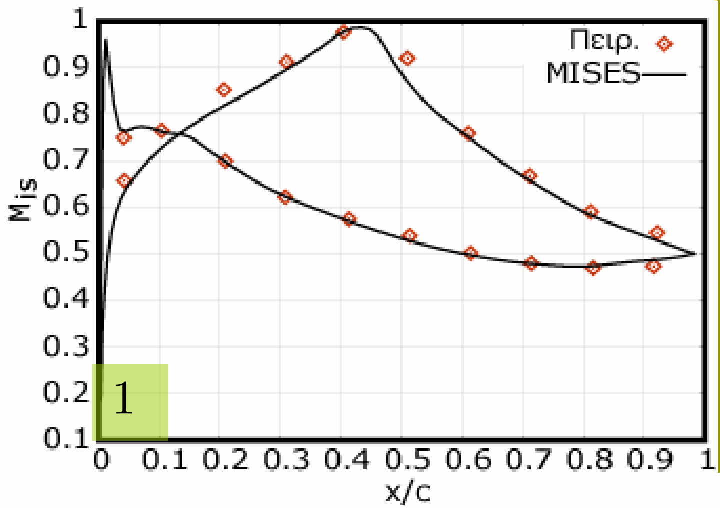


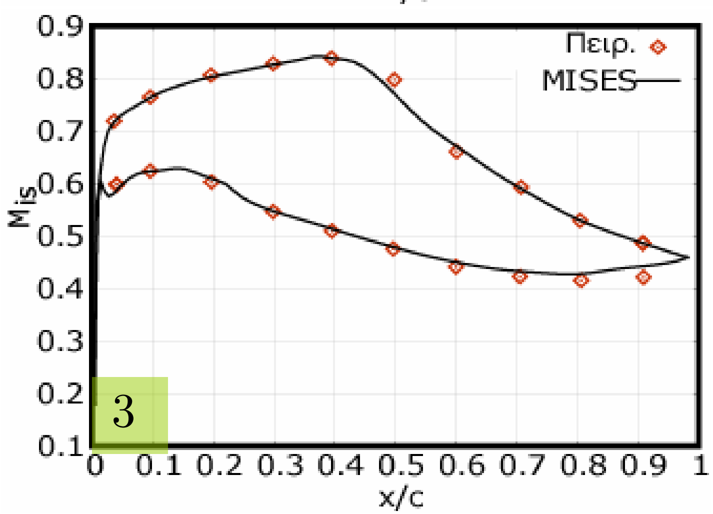
ASME 90-GT-140

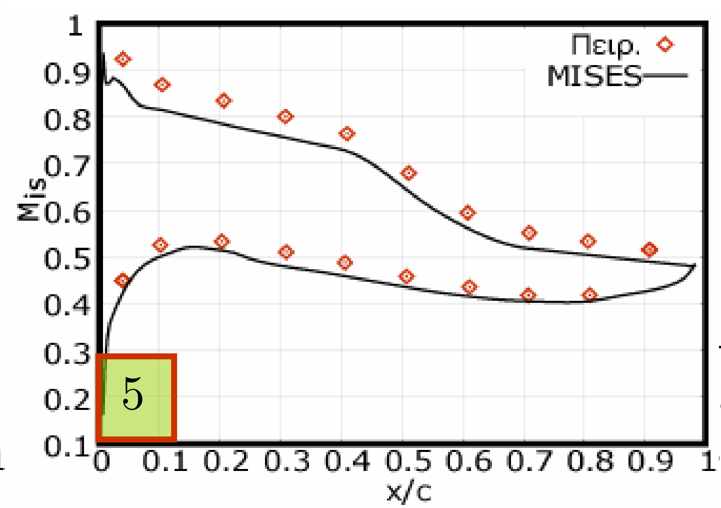
s=0.68c γ=30° c=0.07m

	1	2	3	4	5
α_1	43°	45°	47°	49°	52°
M_1	0.6184	0.6182	0.6180	0.6195	0.6214
Re	8.61E5	8.50E5	8.41E5	8.20E5	7.63E5
1/AVDR	1.0909	1.0965	1.1021	1.1027	1.1032
ω (exp)	0.0232	-	0.0186	-	0.0417
ω MISES	0.0234	0.0208	0.0189	0.0237	0.0374
α_2 (πexp)	20.79	20.80	20.92	21.69	22.74
α_2 MISES	20.80	20.80	20.90	21.70	22.70

MISES 2.53





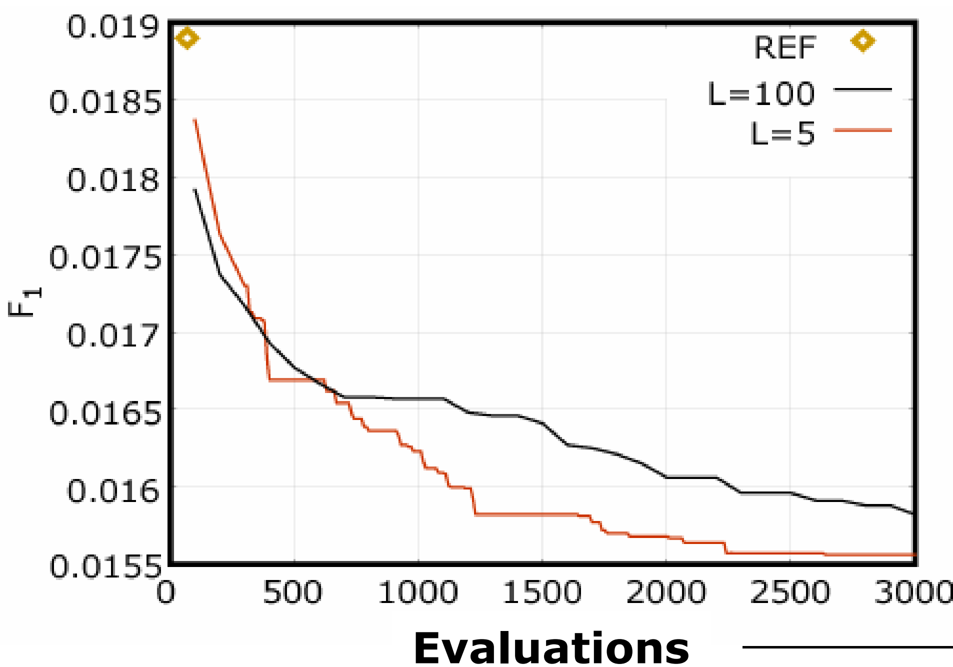


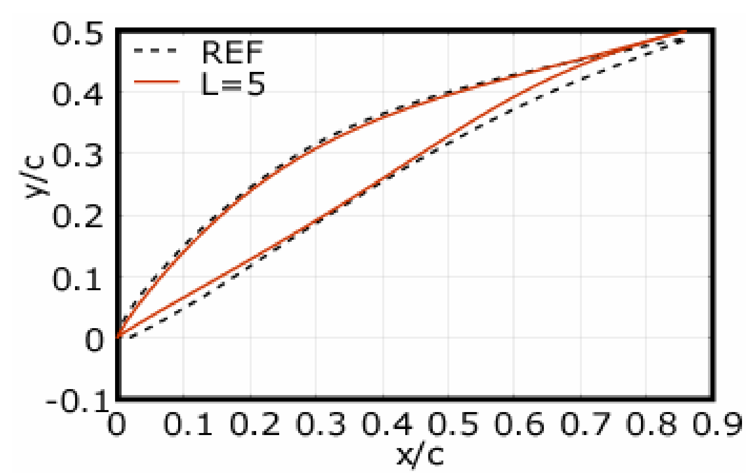
$$\omega = \frac{p_{t2,is} - p_{t2}}{p_{t1} - p_1}$$

Compressor Multi-Point Design (1 OP – 1 target)

(20,2,100)

	1	2	3	4	5
ω L=5	0.0244	0.0182	0.0155	0.0275	-
ω REF	0.0234	0.0208	0.0189	0.0237	0.0374

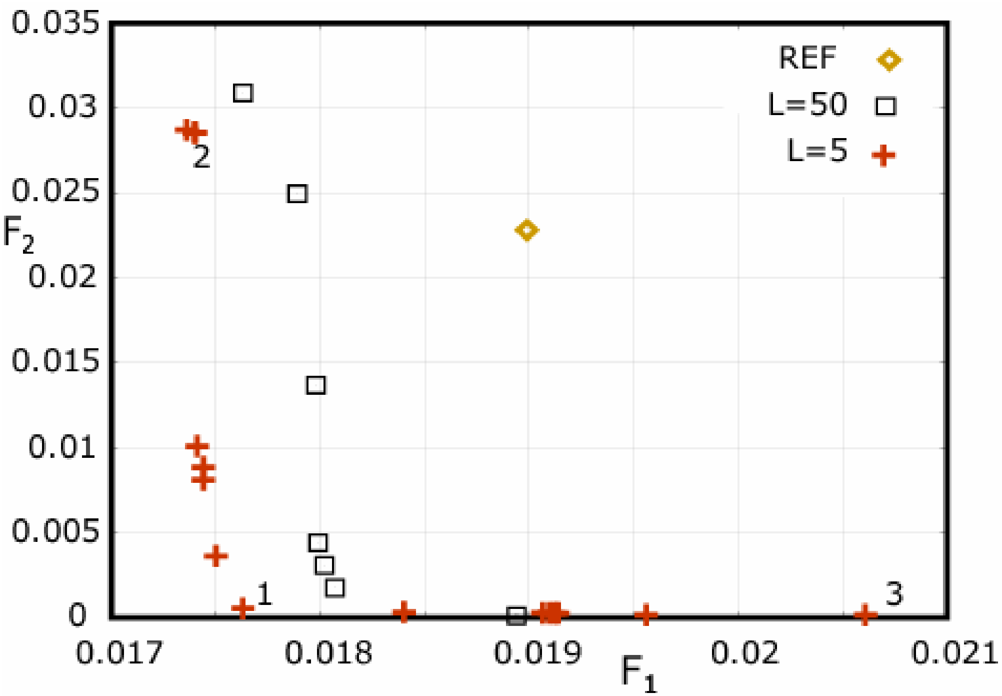


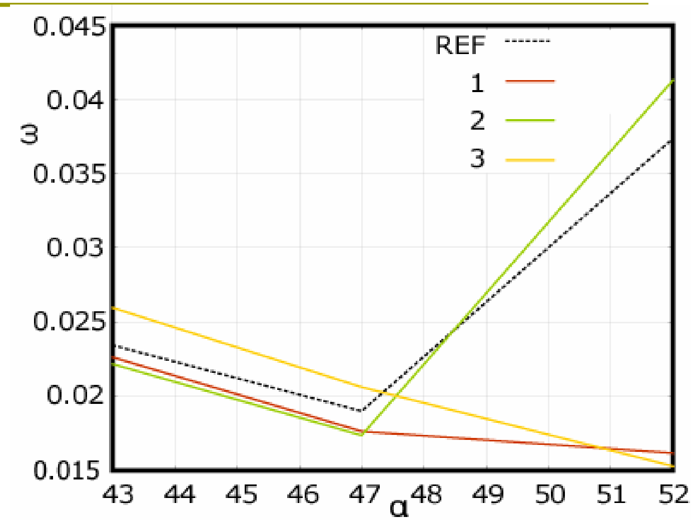


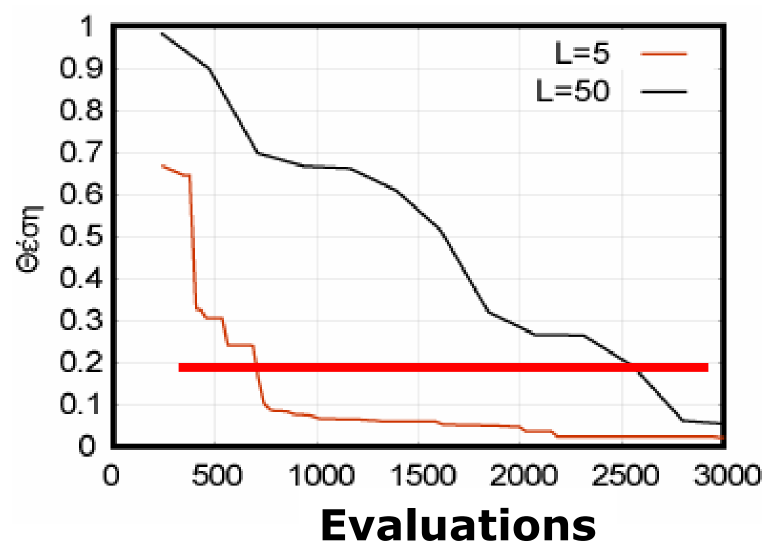
Compressor Multi-Point Design (3 OP – 2 targets)

$$F_1 = \omega_3 \cdot P_1 \cdot P_2$$

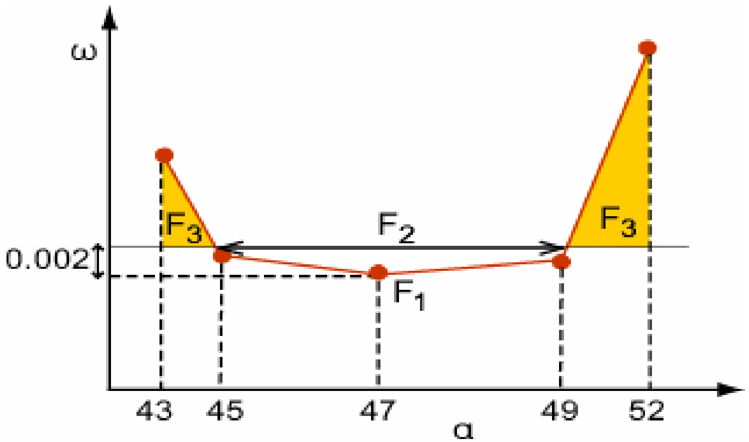
$$F_2 = (\omega_1 + \omega_5 - 2\omega_3) \cdot P_1 \cdot P_2 > 0$$

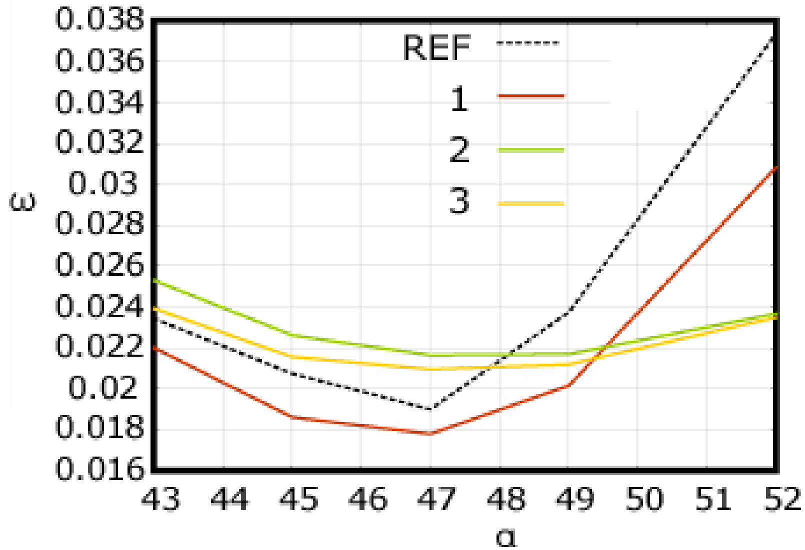


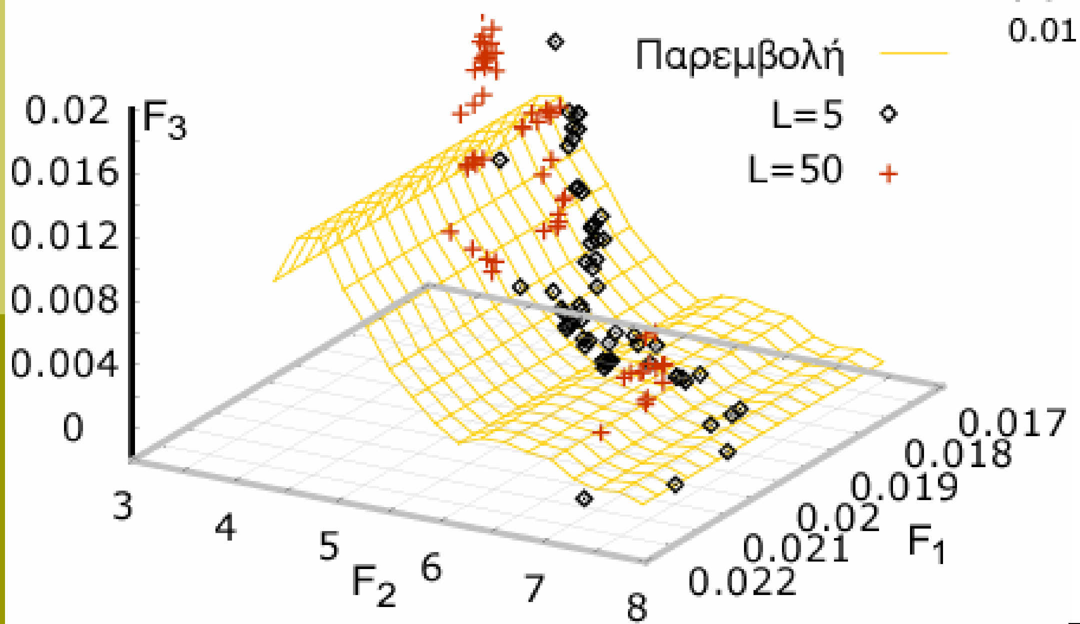


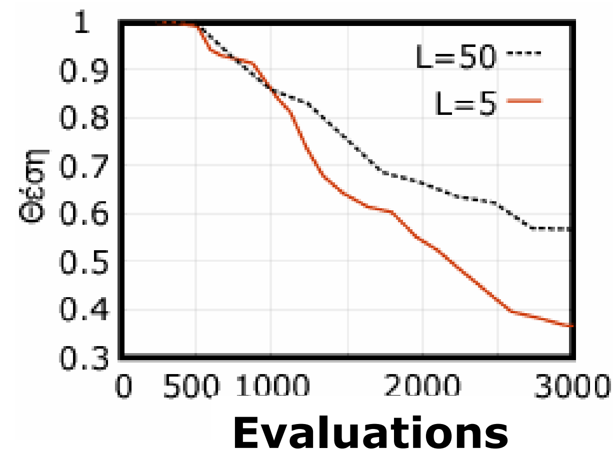


Compressor Multi-Point Design (5 OP – 3 targets)



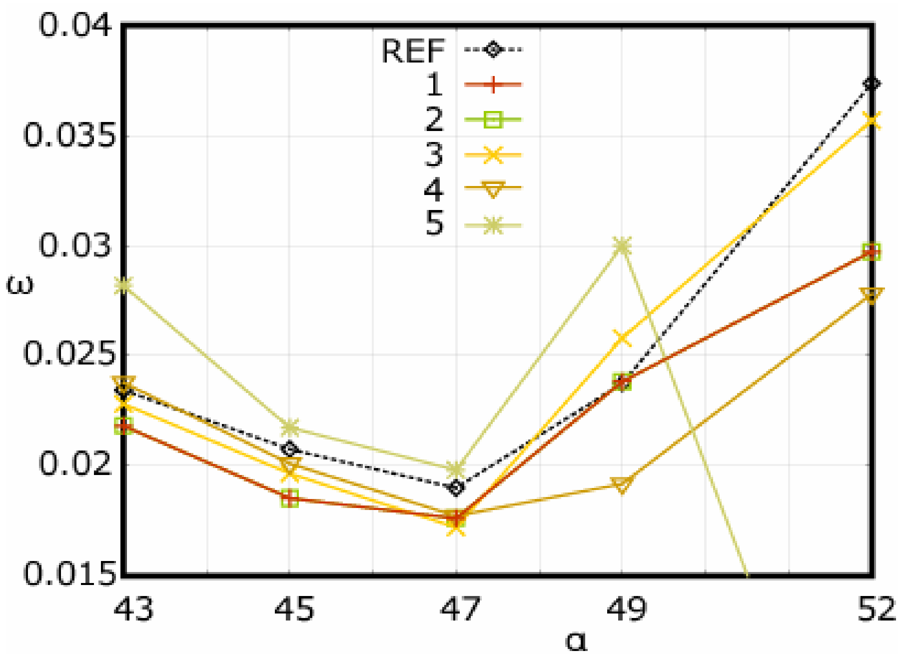


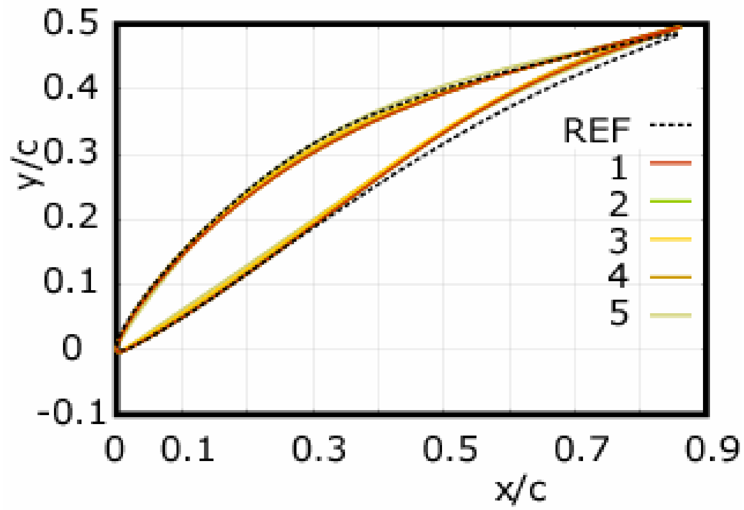


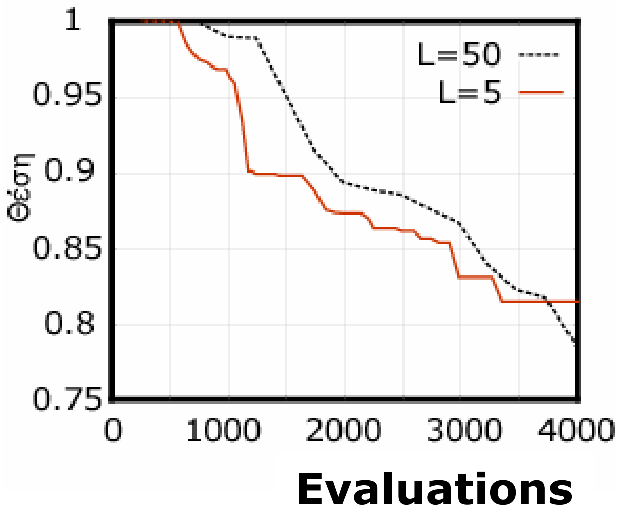


Compressor Multi-Point Design (5 OP – 5 targets)

$$F_i = \omega_i \cdot P_1 \cdot P_2, \;\; i = 1, \ldots, 5$$

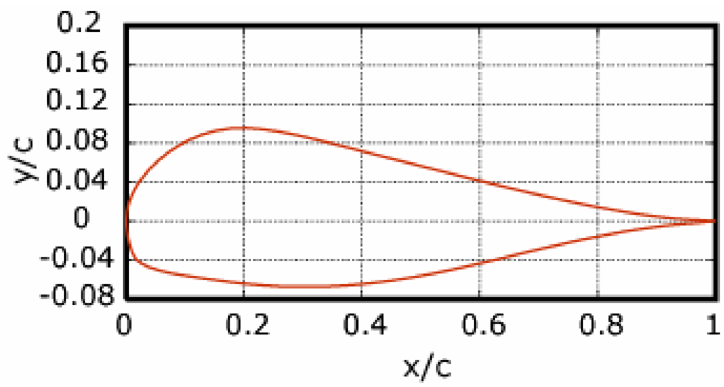


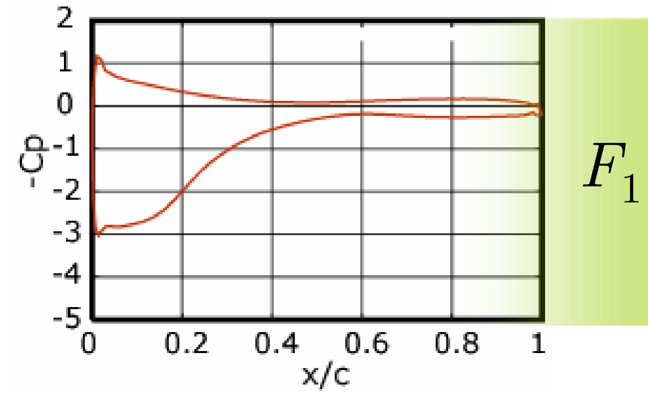




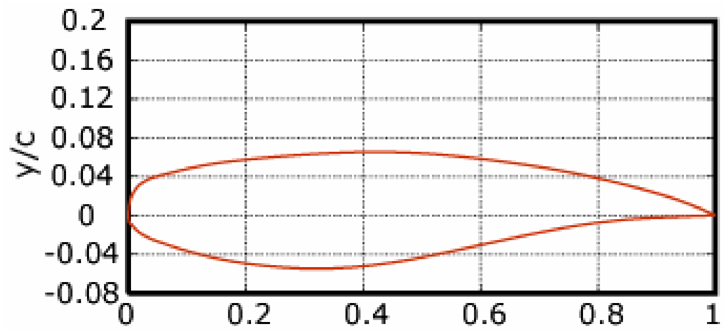
Problem: High-Lift, Low-Drag Optimization

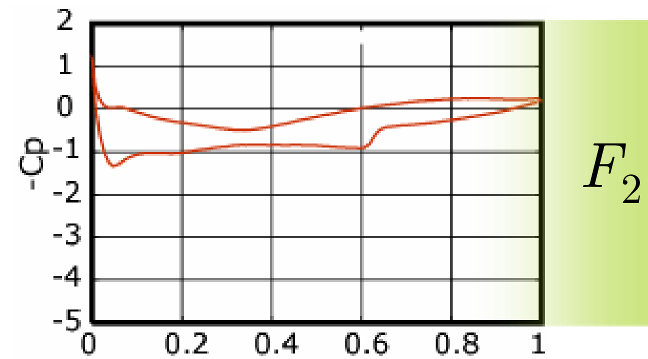
$C_1 : M_\infty = 0.20, \ a_\infty = 10.8^o, \ \mathrm{Re}_\infty = 5 \times 10^6$





$C_2 : M_\infty = 0.77, \ a_\infty = 1.0^o, \ \mathrm{Re}_\infty = 10 \times 10^6$





NS, k-ε WF

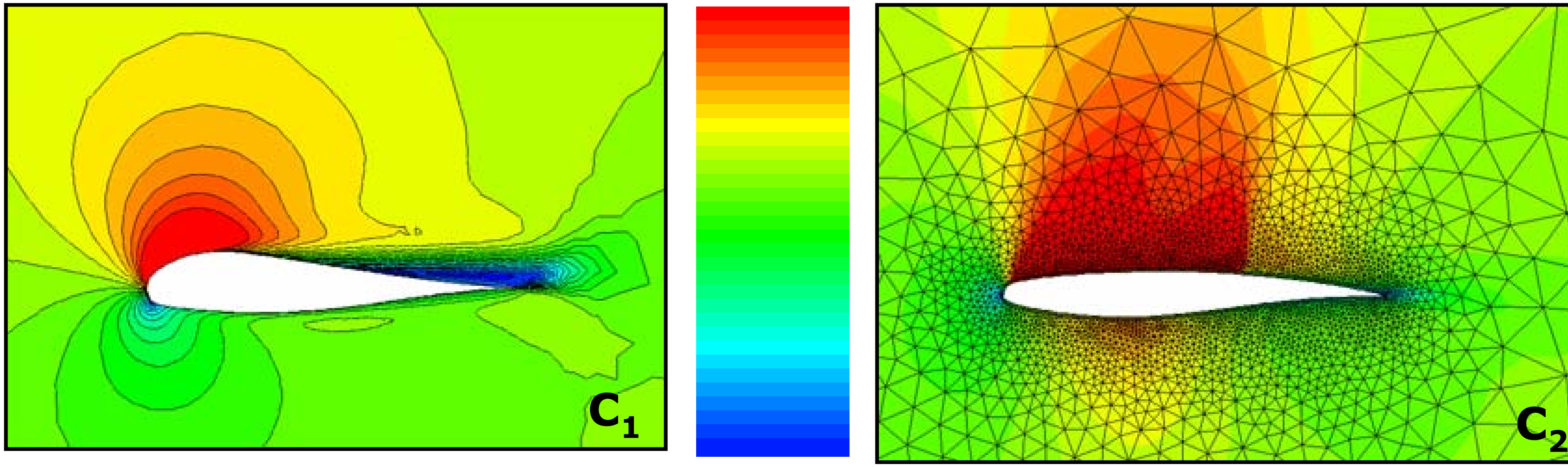
4min

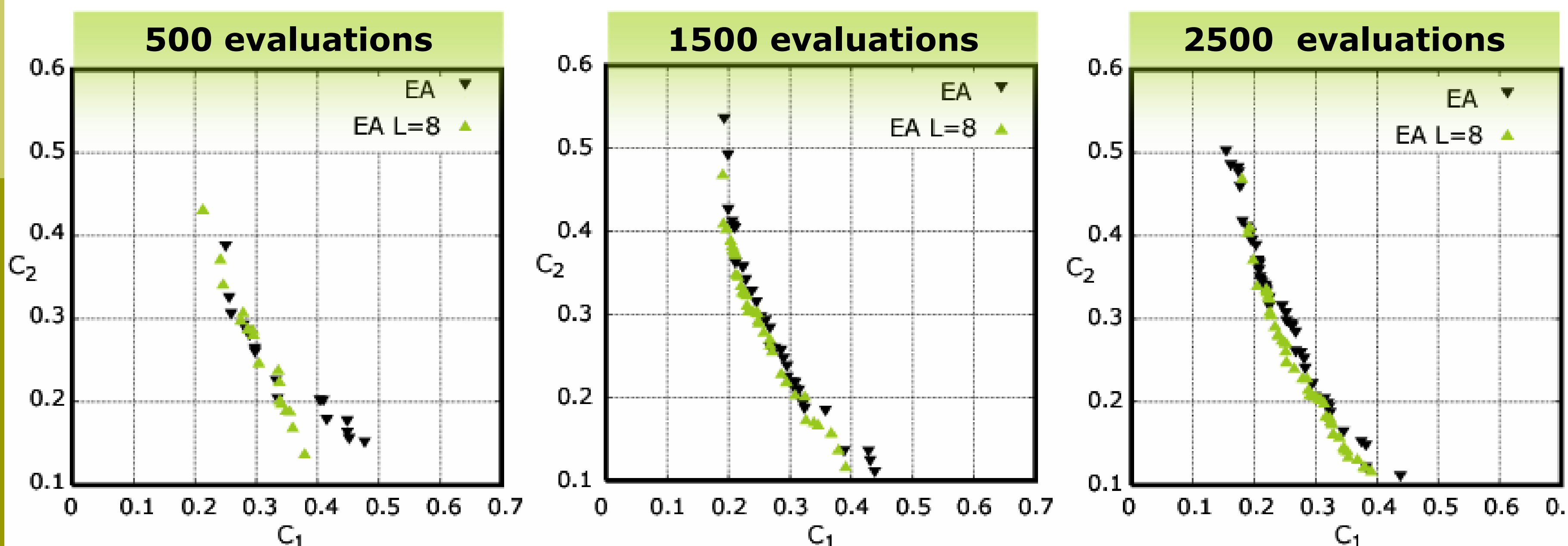
6000

3000

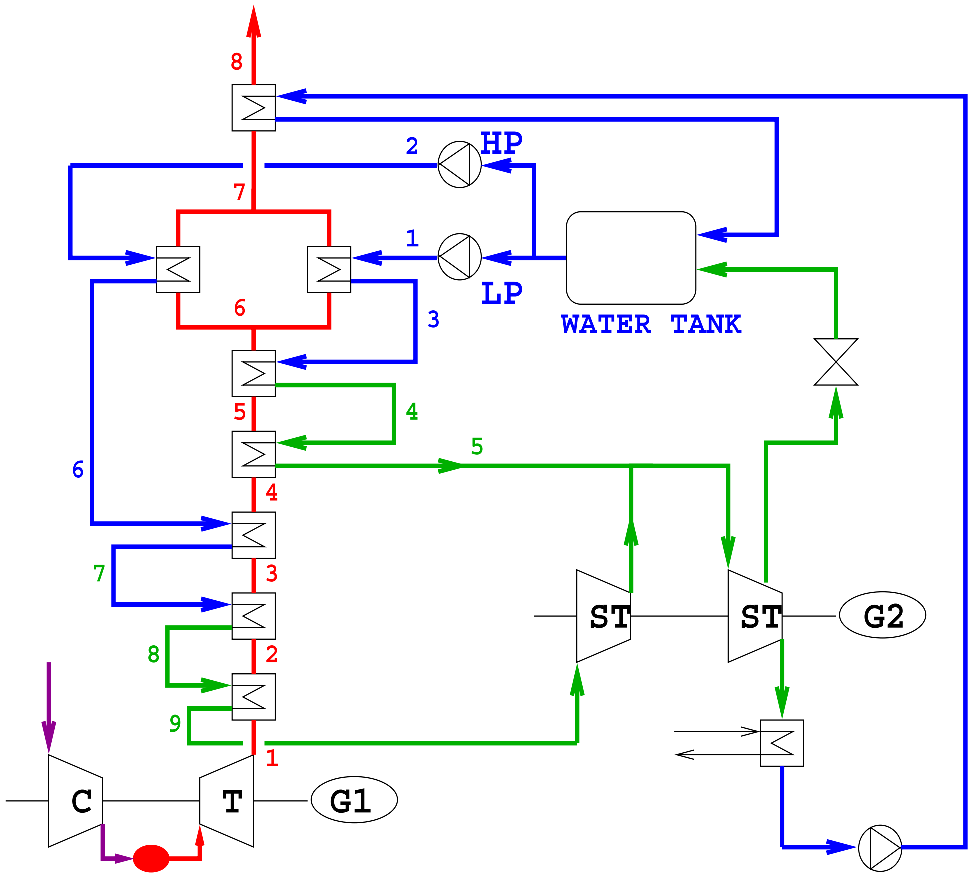
200

Results: High-Lift, Low-Drag Optimization





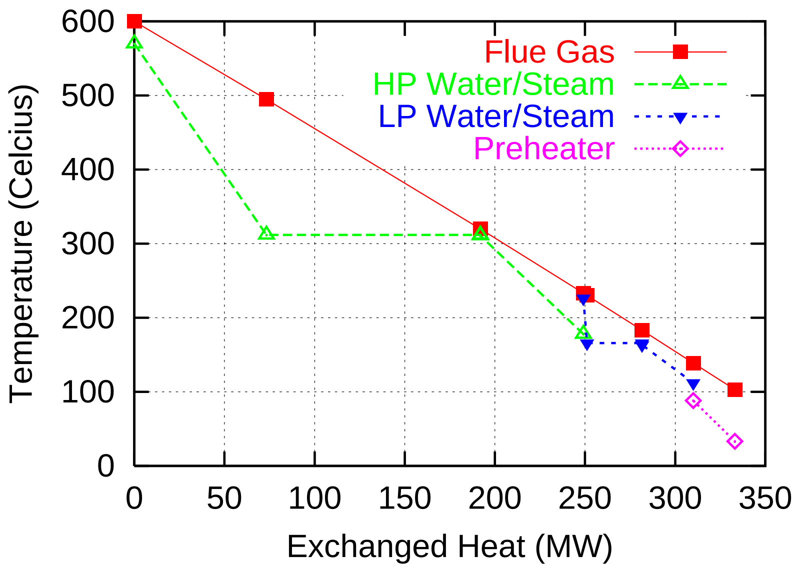
Optimization of Combined Cycle GT Power Plants



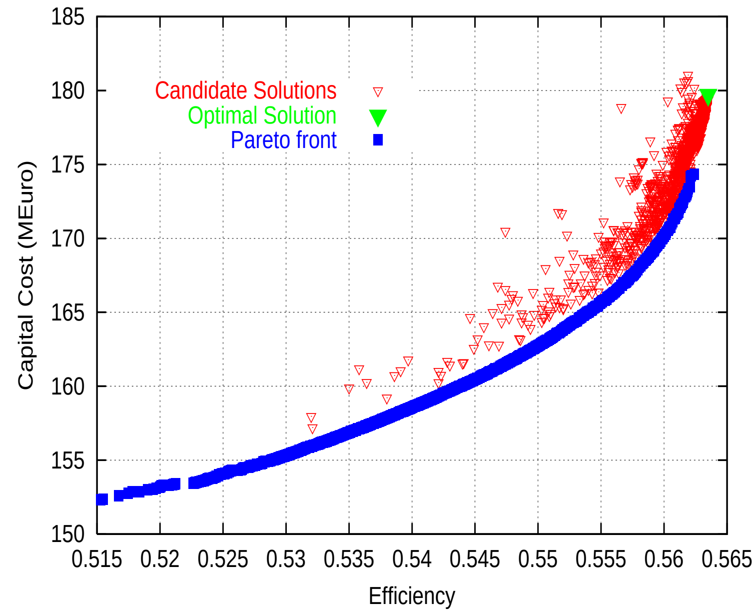
- **Design variables**
  - *HP steam pressure*
  - *LP steam pressure*
  - *superheated HP steam temperature*
  - *feedwater temperature at the inlet to the HP evaporator*
  - *feedwater temperature at the outlet from the first HP economizer*
  - *feedwater temperature at the inlet to the LP evaporator*
  - *superheated LP steam temperature*
  - *steam pressure fed to the water tank*
  - *exhaust gas mass flow ratio (percentage of mass flowrate traversing the LP economizer)*
  - *exhaust gas temperature at the HRSG outlet*
  - *steam extraction pressure from the LP steam turbine*
  - *exhaust gas temperature at the inlet to the condensate preheater*

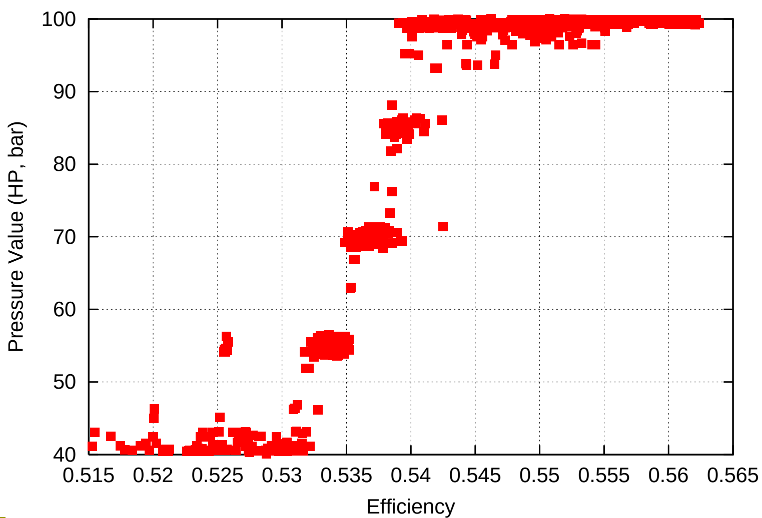
Natural gas fired, dual-pressure CCGTPP configuration
GT: 260 MWe, 38% efficiency, exhaust gas mass flow 615 kg/sec at 600C.

Optimization of Combined Cycle GT Power Plants



Constrained Optimization Problem





HP Pressure Values, over the Pareto Front