NATIONAL TECHNICAL UNIVERSITY OF ATHENS (NTUA)
SCHOOL OF MECHANICAL ENGINEERING
LAB. OF THERMAL TURBOMACHINES
PARALLEL CFD & OPTIMIZATION UNIT (PCOpt/NTUA)

Low-Cost Optimization using Evolutionary Algorithms for Engineering Applications

Kyriakos C. Giannakoglou, Professor NTUA

Dr. V. Asouti, Dr. D. Kapsoulis

Evolutionary Algorithms (EAs), Pros & Cons

EA: **A population-based metaheuristic optimization inspired by the biological evolution. Candidate solutions act as population members for which the value of the cost or fitness function should be computed.**

Pros:
- **Readily accommodates any analysis-evaluation software (as a black-box).**
- **Is gradient-free.**
- **Computes Pareto fronts of non-dominated solutions, in MOO problems.**
- **Handles constraints (for instance, through penalties).**
- **Is amenable to parallelization (simultaneous independent evaluations).**

Cons:
- **Requires a great number of (costly/CFD) evaluations.**

Irrespective of the EA type, methods to reduce the computational cost, without damaging the quality of the optimal solution(s), are needed.

Possible was to Reduce the Computational Cost of EAs

- **Reduce the number of (CFD-based etc., i.e. costly) evaluations:**
  - **Better evolution operators, more careful tuning of the EA.**
  - **Distributed search.**
  - **Replace calls to the costly/exact evaluation tool (PSM, for instance CFD) with calls to cheaper/less-accurate surrogate evaluation models (metamodels).**
  - **Dimensionality reduction (curse of dimensionality).**
  - **Hierarchical/multilevel search.**
  - **Hybridization with gradient-based search, etc.**
- **Reduce the wall-clock time of the optimization:**
  - **Parallelization.**
  - **Asynchronous search by overcoming the generation synchronization barrier.**
- **Combine some (all?) of the above.**

Our EA: The EASY Platform

Without loss in generality, all computations presented in this lecture have been performed using the optimization s/w EASY developed by the PCOpt/NTUA, based on a (μ,λ) EA.

μ = parent population size
λ = offspring population size
e = elite population size

The Evolutionary Algorithm SYstem
http://velos0.ltt.mech.ntua.gr/EASY
http://147.102.55.162/EASY

PCOpt/NTUA provides, free of charge, EASY licenses to academic groups or research institutes, upon request.

Metamodel-Assisted EAs (MAEAs)

The Concept:

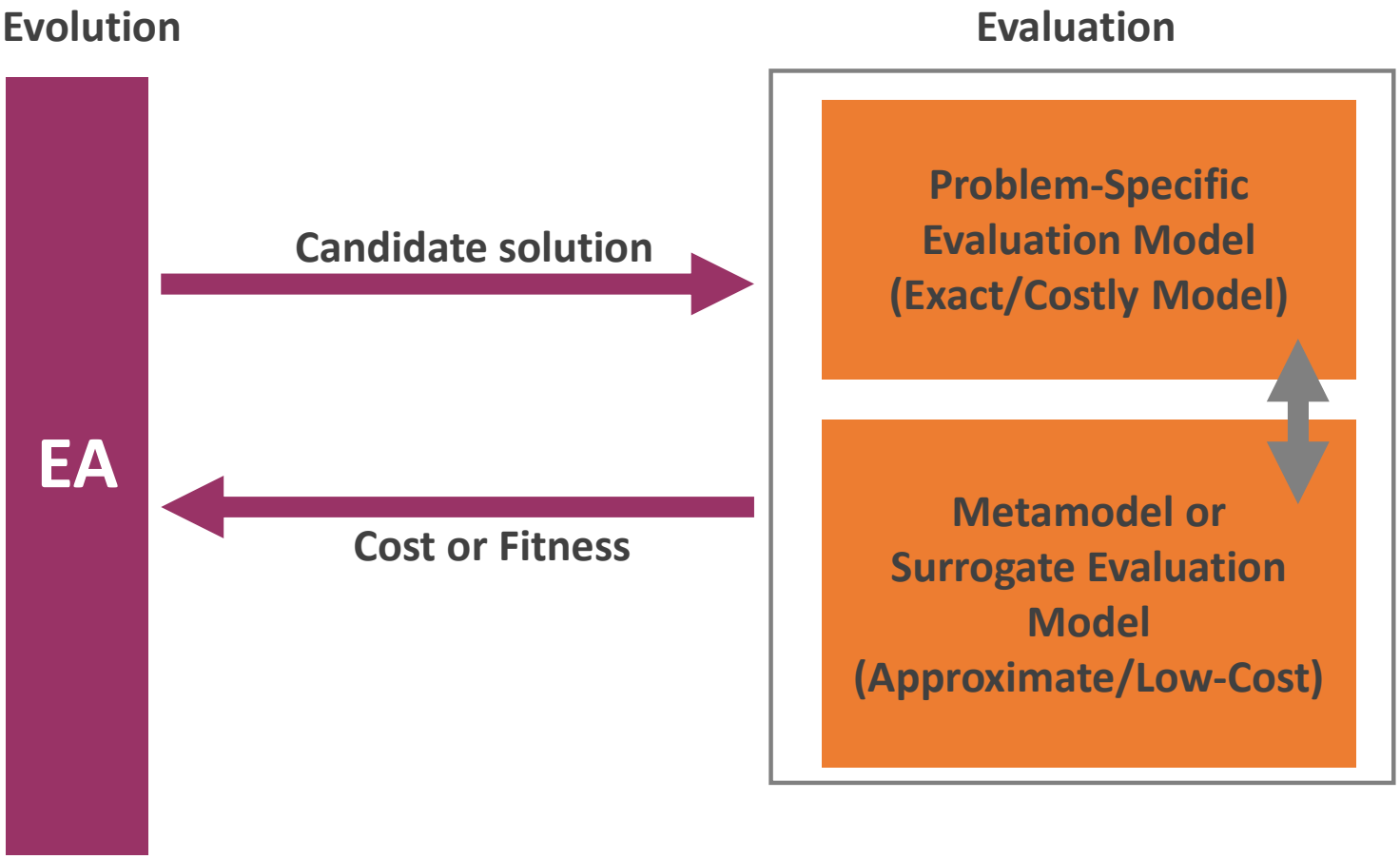
Having already evaluated a number of candidate solutions (individuals), let us train and use an "interpolation" method (a model-agnostic black-box: metamodel or surrogate evaluation model) to approximate the objective or constraint function values of new individuals generated by the EA, by avoiding the use of the costly Problem Specific Model (PSM; here, the CFD code!)

Valid for either Single- & Multi-Objective Optimization (SOO & MOO)

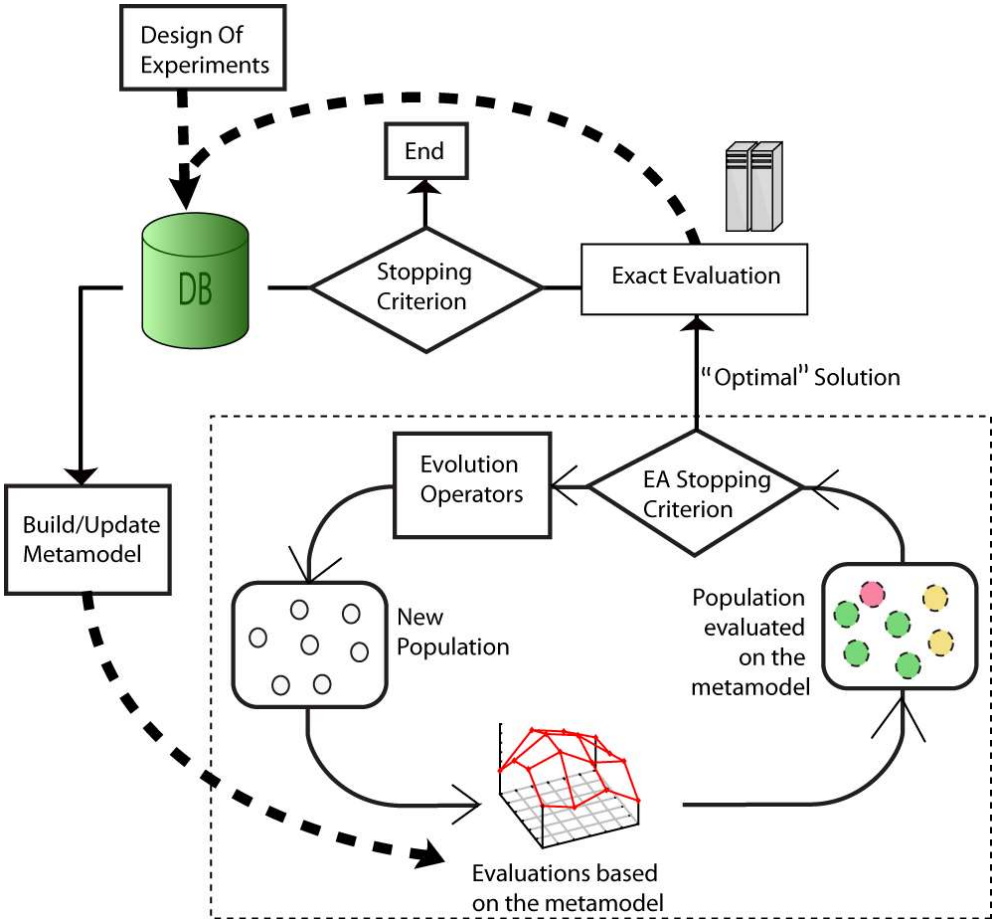
Questions:

- **When and how to train the metamodel?**
- **Which metamodel (polynomial regression, neural networks, …)?**
- **How to collect training samples for the metamodel?**
- **How to use the cost/fitness function approximation provided by the metamodel?**

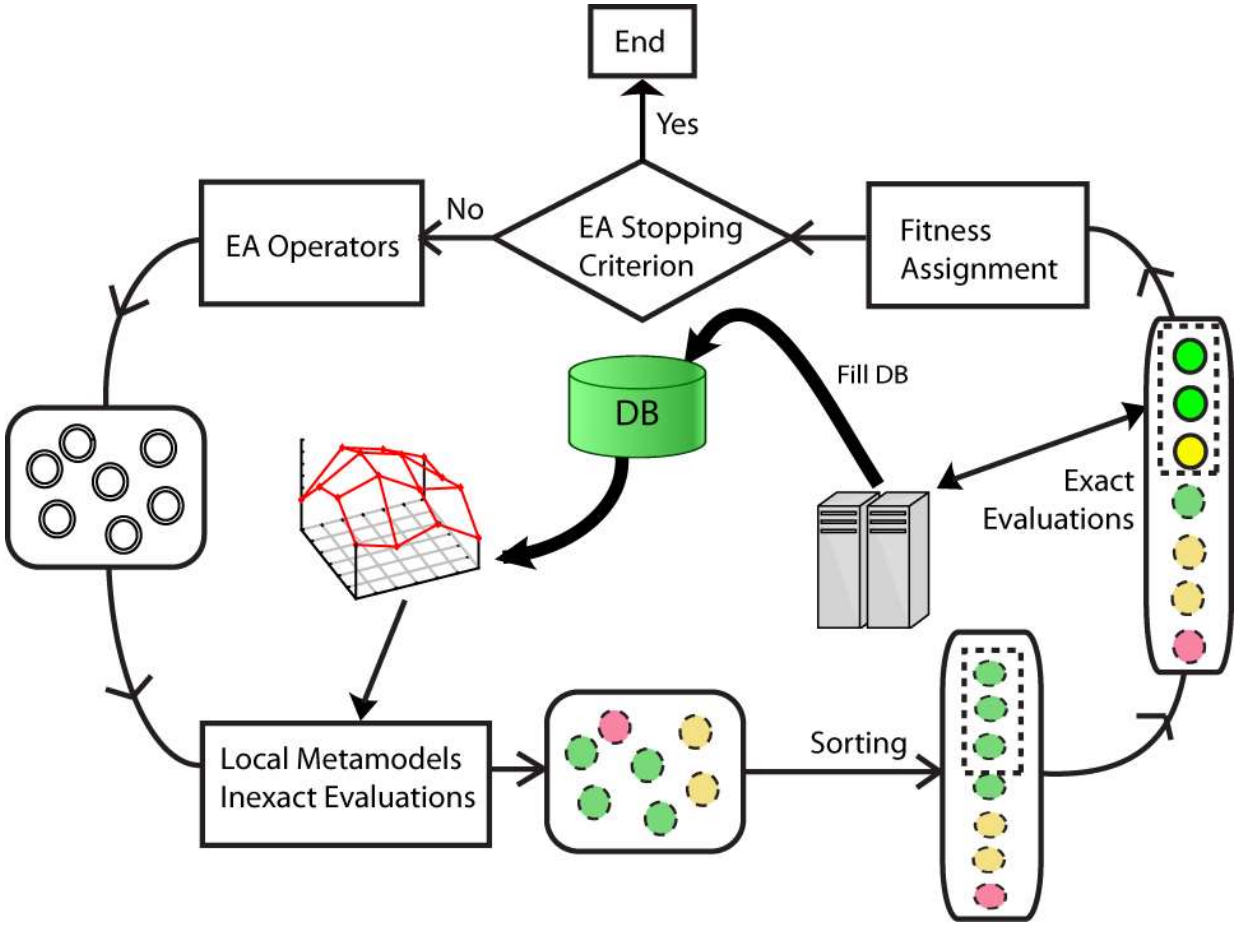
Metamodel-Assisted EAs (MAEAs)



Metamodel-Assisted EAs (MAEAs) with off-line Trained Metamodels



Metamodel-Assisted EAs (MAEAs) with on-line Trained Metamodels



The Inexact Pre-Evaluation (IPE) Technique

Low-Cost Pre-Evaluation (LCPE) Technique

MAEAs: Ways to Implement Metamodels

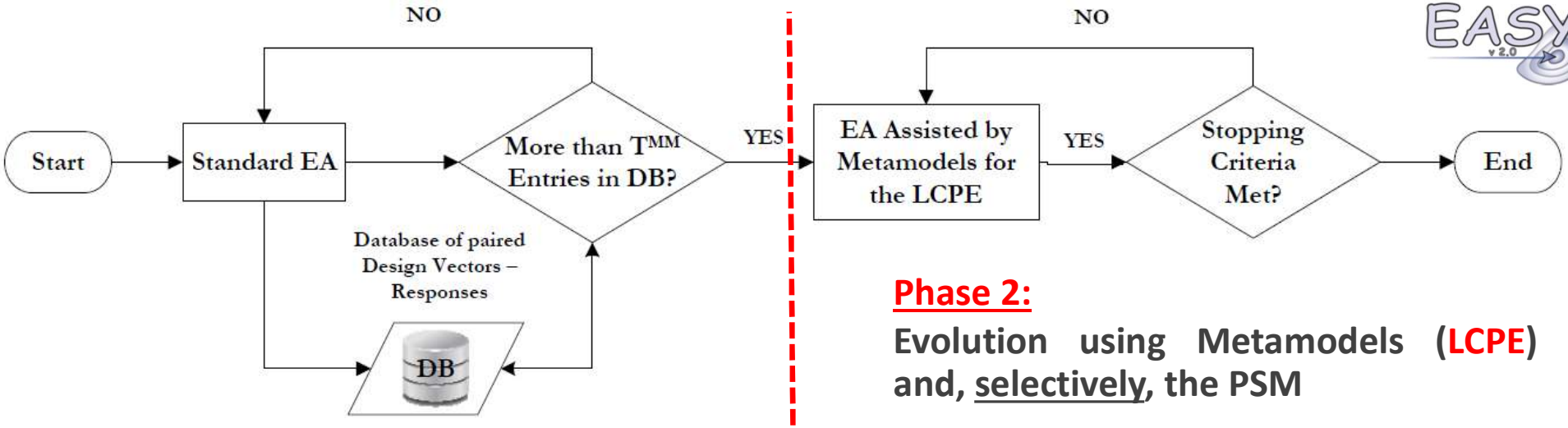
MAEAs with Off-Line Trained Metamodels:

- A Design of Experiment (DoE) technique is used to sample the design space, collect training patterns, evaluate them on the PSM & store them into the Database (DB).
- A (global) metamodel is trained and the EA search relies exclusively upon its use.
- "Optimal" solution(s) is/are re-evaluated on the PSM (e.g. the CFD code).
- New samples are collected & evaluated on the PSM, DB is enriched, a new (hopefully, better) metamodel is trained…
- Iterate (successive EA-based searches) until convergence.

MAEAs with On-Line Trained Metamodels:

- Local metamodels are trained during the evolution by on-line collected training patterns. The Low-Cost Pre-Evaluation (LCPE) phase.
- Coordinated use of metamodels and the PSM.
- Considered to be the distinguishing feature of EASY .

A MAEA with On-Line Trained Metamodels



Phase 1:

Evolution based exclusively on the PSM; no use of Metamodels

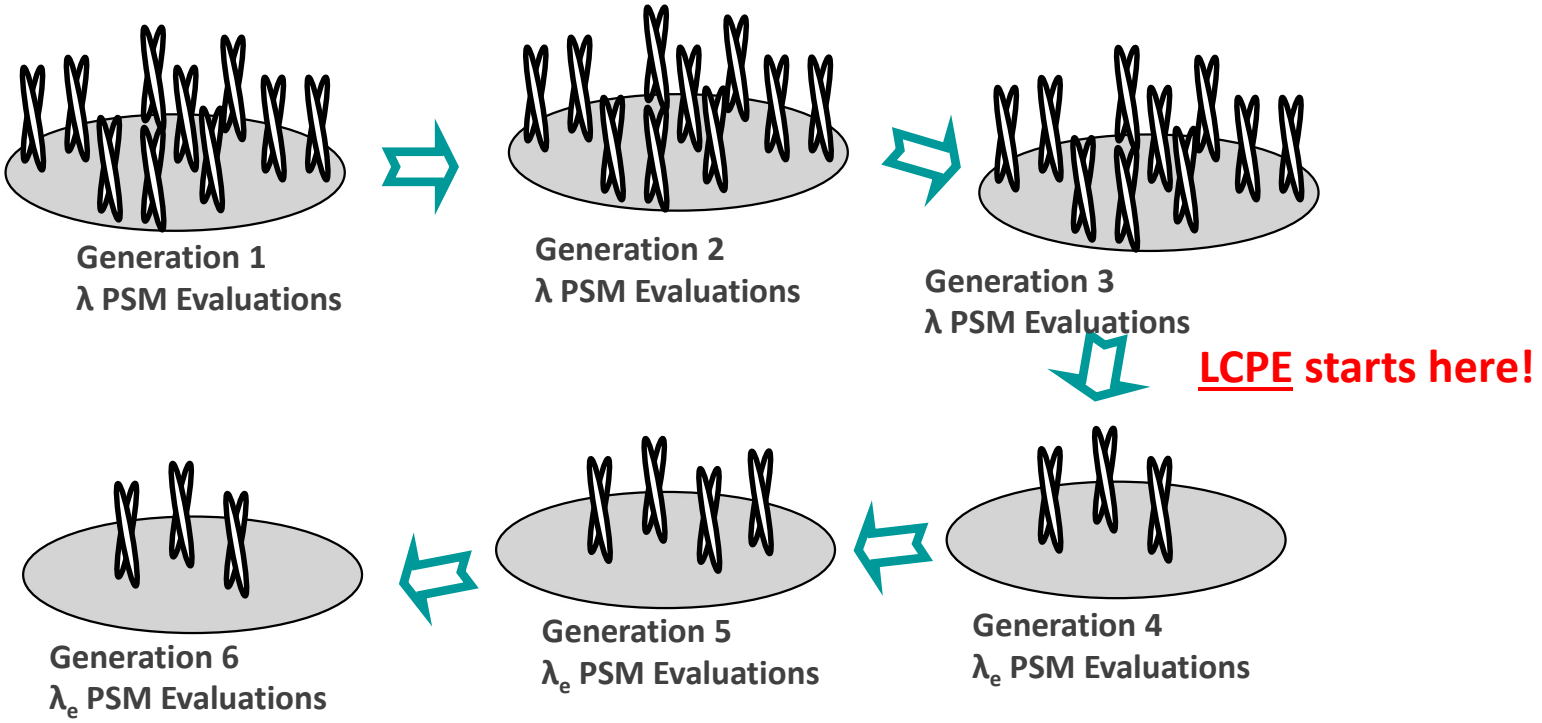
Upon completion, the DB has at least T^{MM} inputs

Phase 2:

Evolution using Metamodels (LCPE) and, selectively, the PSM

- **The use of metamodels starts once T^{MM} evaluated (on the PSM) individuals exist in the DB.**
- **All population members are evaluated on local/personalized metamodels trained on neighboring data in the DB.**
- **The best λ_e(«λ, $\lambda_{e,min} \leq \lambda_e \leq \lambda_{e,max}$) of them are re-evaluated on the PSM & stored in the DB; this determines the computational cost of this generation.**

A MAEA with On-Line Trained Metamodels – The LCPE Phase

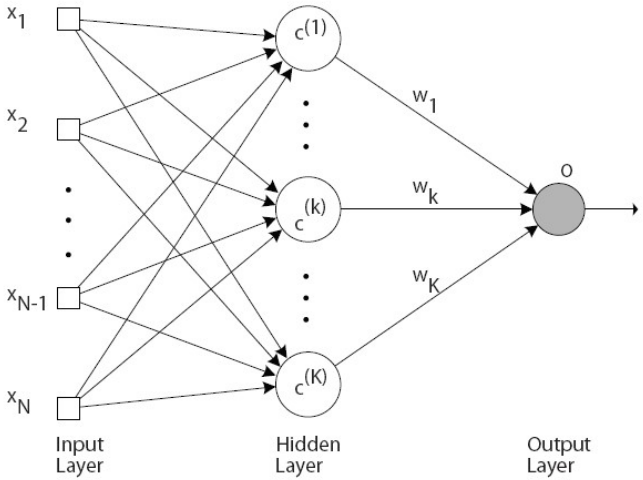


Int. Review Journal Progress in Aerospace Sciences, 38:43-76, 2002.

On Metamodels (RBF Networks)

The Radial Basis Function (RBF) Network:

Without loss of generality, they are exclusively used in all presented studies.



RBF network with N inputs,
K hidden units (RBF centers) and
a single output (M_o=1).
To be trained with T training patterns.

$$o(\vec{x}) = \sum_{k=1}^{K} w_k \mathcal{G}(\|\vec{x} - \vec{c}^{(k)}\|_2)$$

Possible activation functions:

$$\mathcal{G}(h) = exp(-h^2/r^2) \qquad \mathcal{G}(h) = (h^2 + r^2)^{1/2}$$

$$\mathcal{G}(h) = (h^2 + r^2)^{-1/2} \qquad \text{etc.}$$

Training:

$$\sum_{t=1}^{T} w_t \mathcal{G}(\|\vec{x}^{(t)} - \vec{c}^{(t)}\|_2) = \zeta^{(t)}$$

Interpolation (K=T) or Approximation (K<T)?

On Metamodels (RBF Networks)

The Radial Basis Function (RBF) Network - Training:

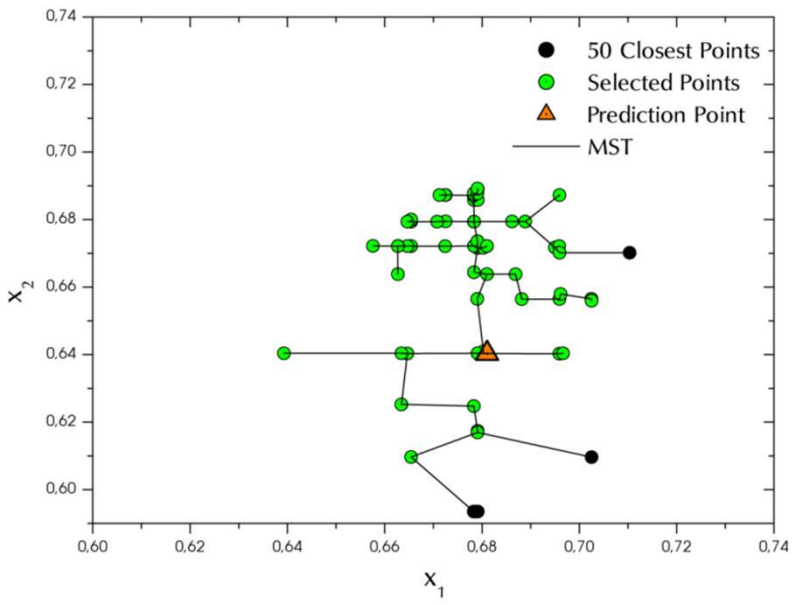
Selection of the RBF centers (coincide with the training patterns) and solution of the linear/symmetric system (case K=T; interpolation):

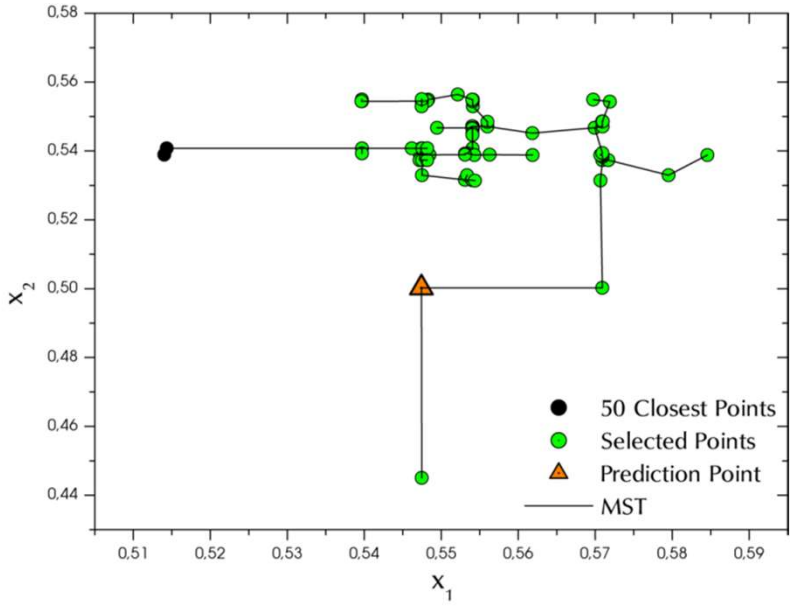
$$\begin{bmatrix} \mathcal{G}(\|\vec{x}^{(1)} - \vec{c}^{(1)}\|_2) & \cdots & \mathcal{G}(\|\vec{x}^{(1)} - \vec{c}^{(T)}\|_2) \\ \vdots & \ddots & \vdots \\ \mathcal{G}(\|\vec{x}^{(T)} - \vec{c}^{(1)}\|_2) & \cdots & \mathcal{G}(\|\vec{x}^{(T)} - \vec{c}^{(T)}\|_2) \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_T \end{bmatrix} = \begin{bmatrix} \zeta_1 \\ \vdots \\ \zeta_T \end{bmatrix}$$

On Metamodels (RBF Networks)

The Radial Basis Function (RBF) Network – Selection of Training Patterns:

Building the minimum spanning tree in the design space, starting from the new individual.
Identification & special treatment of outliers.





On Metamodels (RBF Networks)

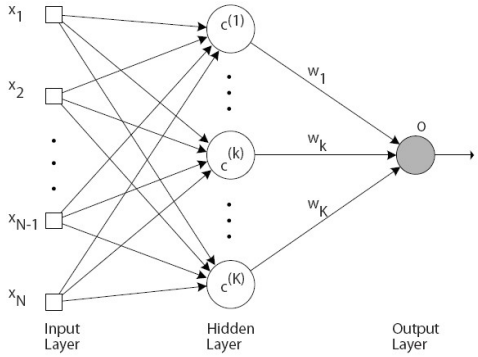
The Radial Basis Function (RBF) Network – Selection of RBF Centers:

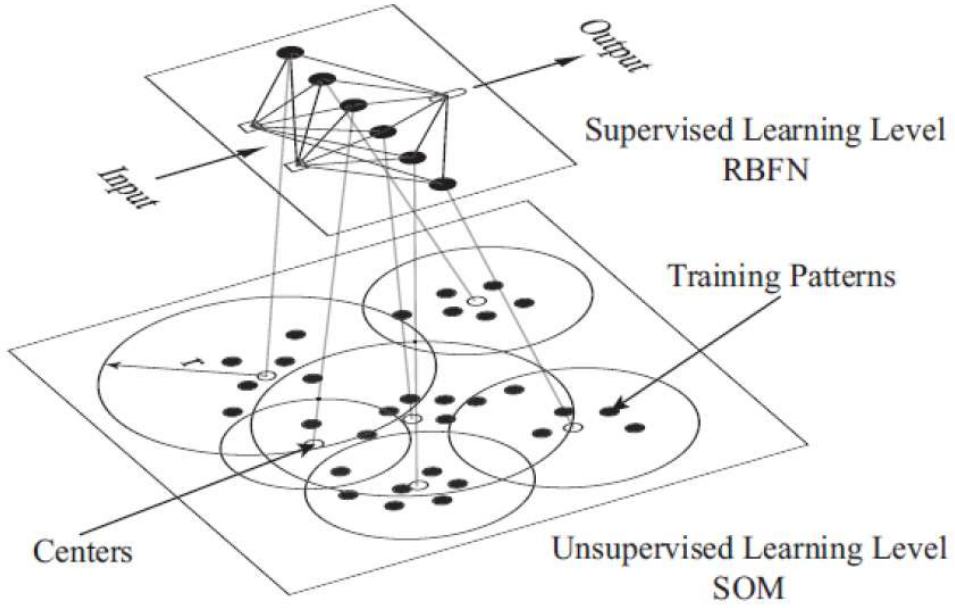
Increase network's generalization by using less hidden nodes than training patterns (K<T).

Selection of RBF centers using Self-Organized Maps (SOMs).

Two levels of training:

- **Unsupervised (classification of training patterns in K clusters).**
- **Supervised (training by minimizing the approximation error).**





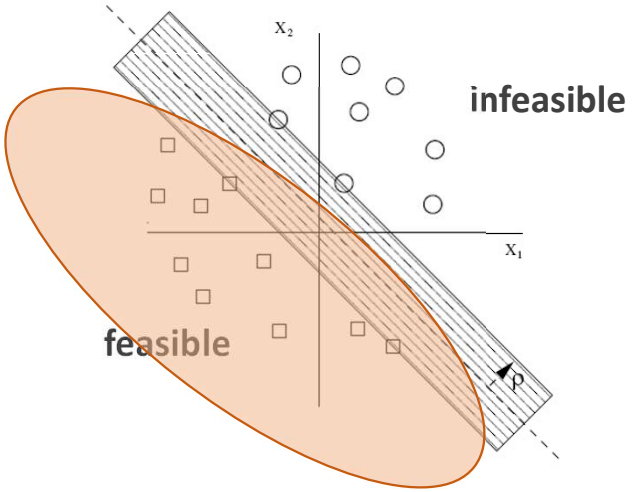
Metamodels in Constrained Optimization

Constraint Handling in the LCPE through Support Vector Machines (SVM):

Question: Since infeasible population members are penalized (with a +/- infinite F value), should we include them in the training patterns for the metamodel or not?

Possible remedy: Before the use of the RBF network in the LCPE phase of the MAEA, train/use an SVM to guess whether the new individual is feasible or infeasible:

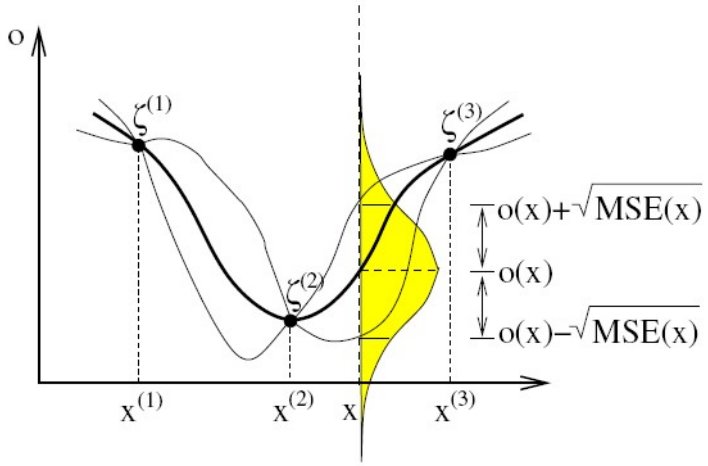
- **RBF is used only for individuals marked as "feasible".**
- **Individuals marked as "infeasible" are penalized.**



Other Metamodels

Kriging:

Estimate and use (as an extra criterion) the Confidence Interval which is pertinent to the guessed/approximated value of the objective or constraint function.



IEEE Transactions on Evolutionary Computation, 10:421-439, 2006.

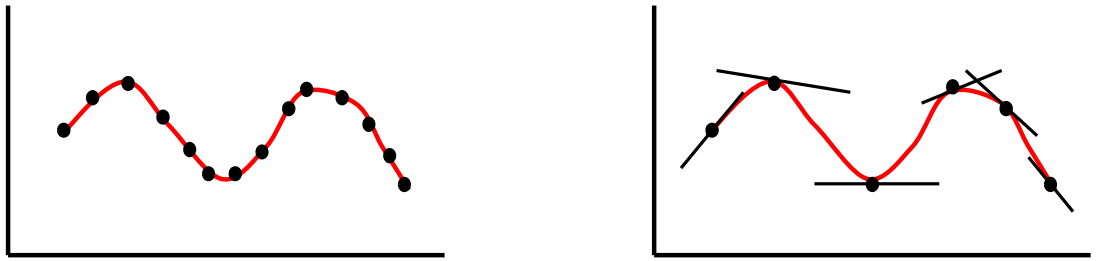
Other "Enhanced" Metamodels

Gradient-Assisted RBF Network:

Prerequisite: The availability of a method to compute the gradient of the objective function (such as the adjoint method in CFD).

Less training patterns, better approximations.

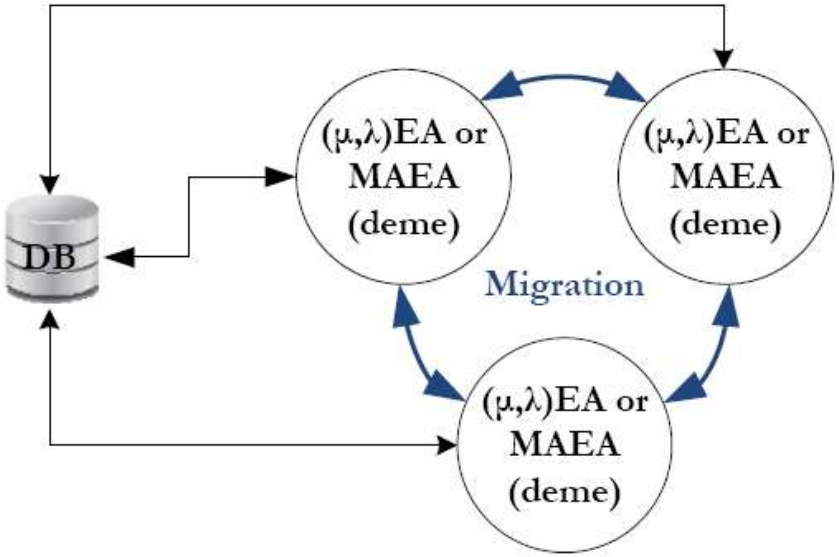
Inspired by Lagrange vs. Hermite interpolation.



- *Applied Mathematical Modeling, 28:197-209, 2004.*
- *Computer Methods in Applied Mechanics and Engineering, 195:6312-6329, 2006.*
- *Journal of Inverse Problems in Science & Engineering, Vol. 14(4):397-410, 2006.*

Distributed EAs and MAEAs

Run more than one EAs, each one with its own parent & offspring populations, in semi-isolation: by regularly exchanging promising etc individuals.



User-Defined Parameters:

- **Number of demes or islands**
- **Communication topology**
- **Communication frequency**
- **Migration policy**
- **EA set-up per deme; exploration/exploitation oriented demes!**

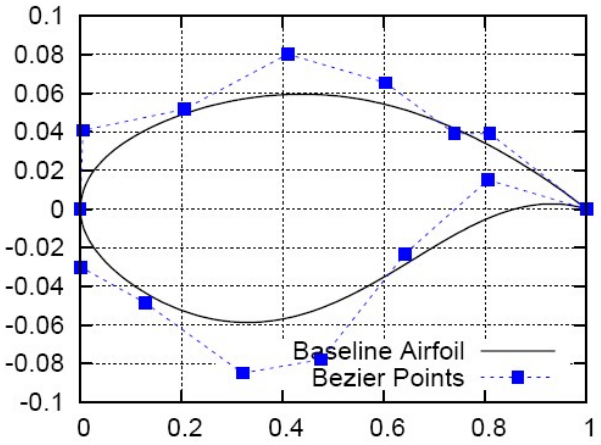
Common DB for all demes.

International Journal for Numerical Methods in Fluids, 53:455-469, 2007.

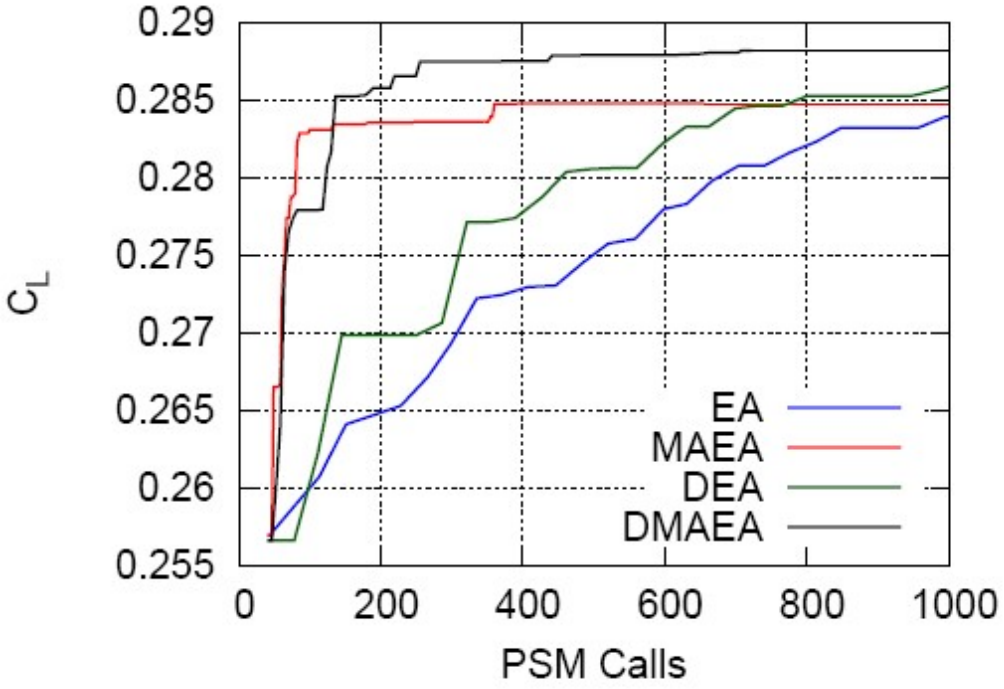
Description of Demo Cases

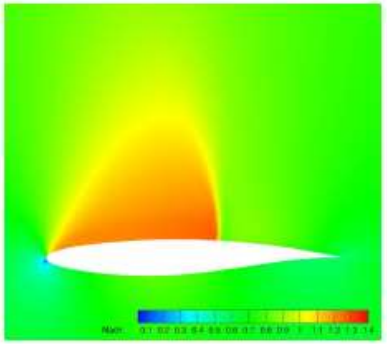
Demo Case 1 (SOO):

Shape optimization of an isolated airfoil.
Target: max. Lift (C_L).
Inviscid flow, M_{inf}=0.40, a_{inf}=5°.
Constraints: Lower and upper bounds of the design variables.
Unstructured grid, ~20K nodes.
Parameterization: 2 Bezier curves (8 Control Points each).
N=12 DoFs (internal control points moving in the y-direction).
PSM: The PCOpt/NTUA GPU-enabled flow solver (PUMA).
Cost per evaluation: ~ 5 sec. on one NVIDIA K20 GPU.
Basis of Comparison: a (20,40)EA.

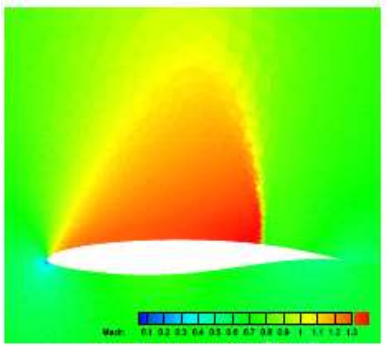


Comparisons on Demo Case 1





Starting



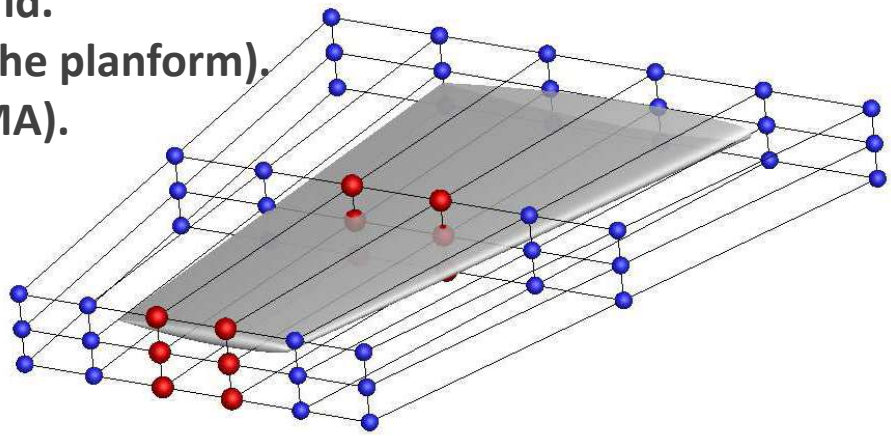
Optimal

(20,40) EA
(20,40) MAEA, T^{MM}=40, λ_e=3
DEA or DMAEA: two demes (10,20)

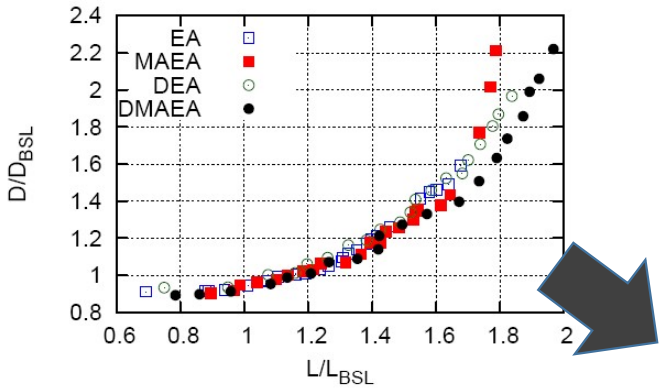
Description of Demo Cases

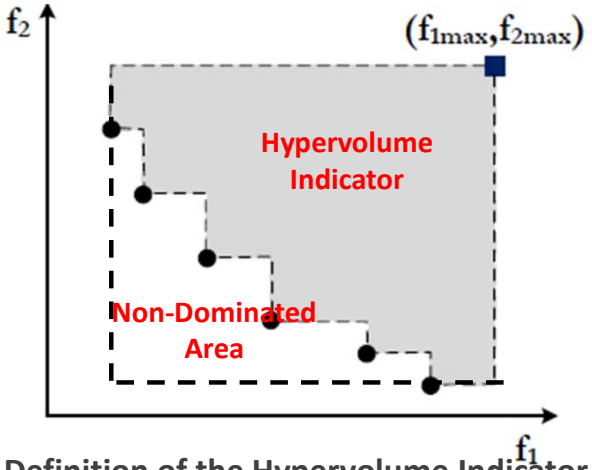
Demo Case 2 (MOO):

Shape optimization of an isolated wing (two objectives)
Target: max. Lift (L) and min. Drag (D)
Inviscid flow, M_{inf}=0.40, $a_{pitch,inf}$=3.06°, $a_{yaw,inf}$=0°.
Constraints: Lower and upper bounds of the design variables.
Unstructured grid, ~1.33Mi nodes.
Parameterization: 6x3x3 Volumetric NURBS Control Grid.
N=24 DoFs (internal control points moving normal to the planform).
PSM: The PCOpt/NTUA GPU-enabled flow solver (PUMA).
Cost per evaluation: ~ 2 min. on one NVIDIA K20 GPU.
Basis of Comparison: a (10,20)EA.

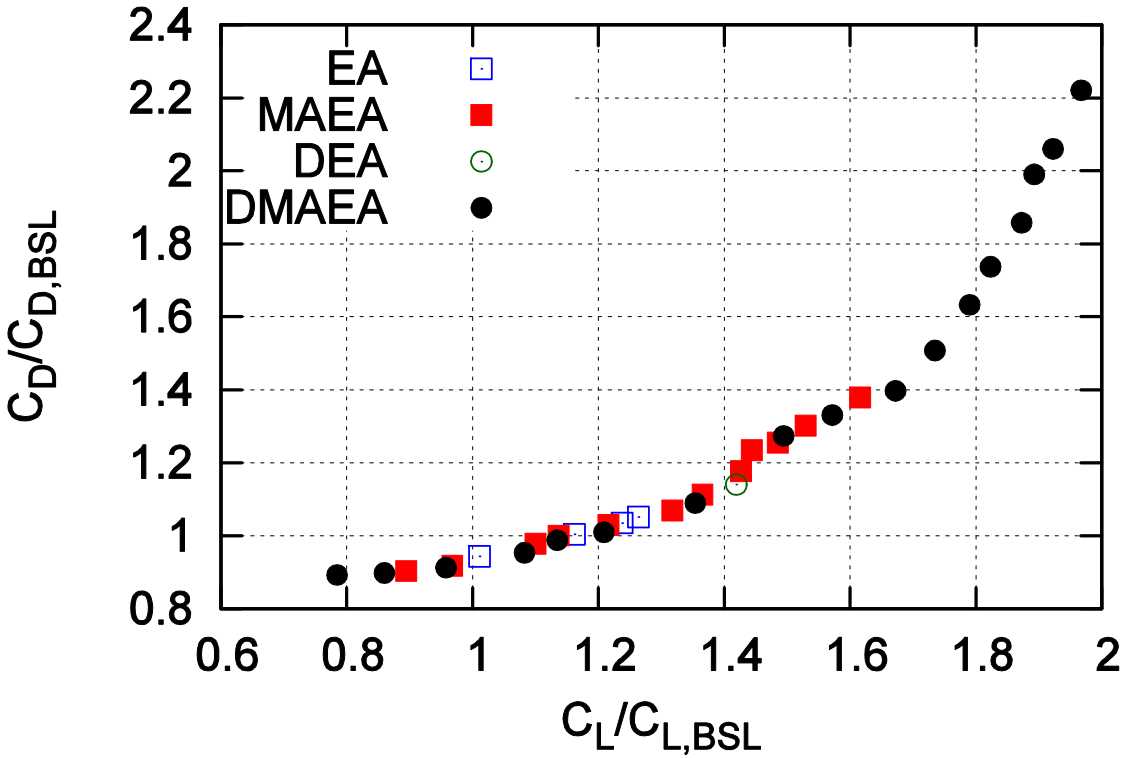


Comparisons on Demo Case 2



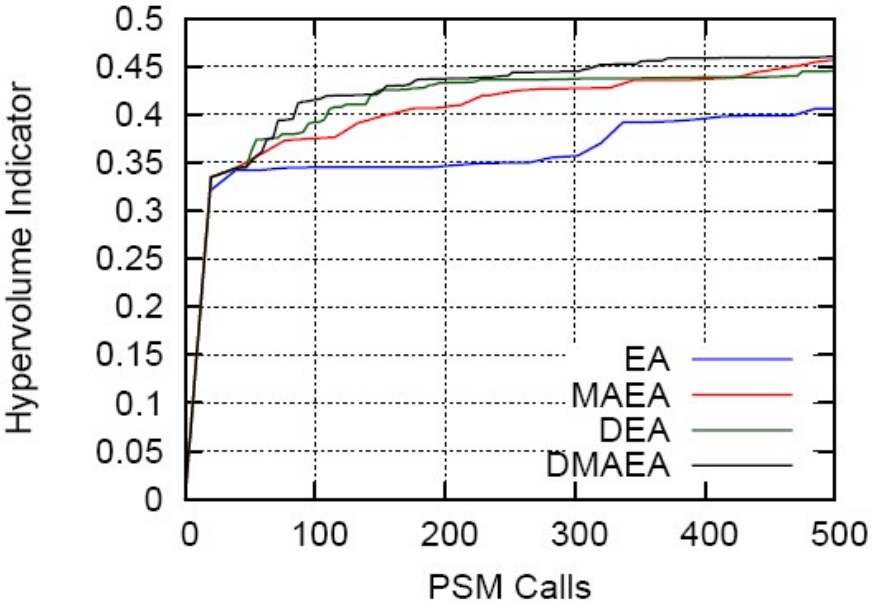


Definition of the Hypervolume Indicator (for min. f_1 & min. f_2).

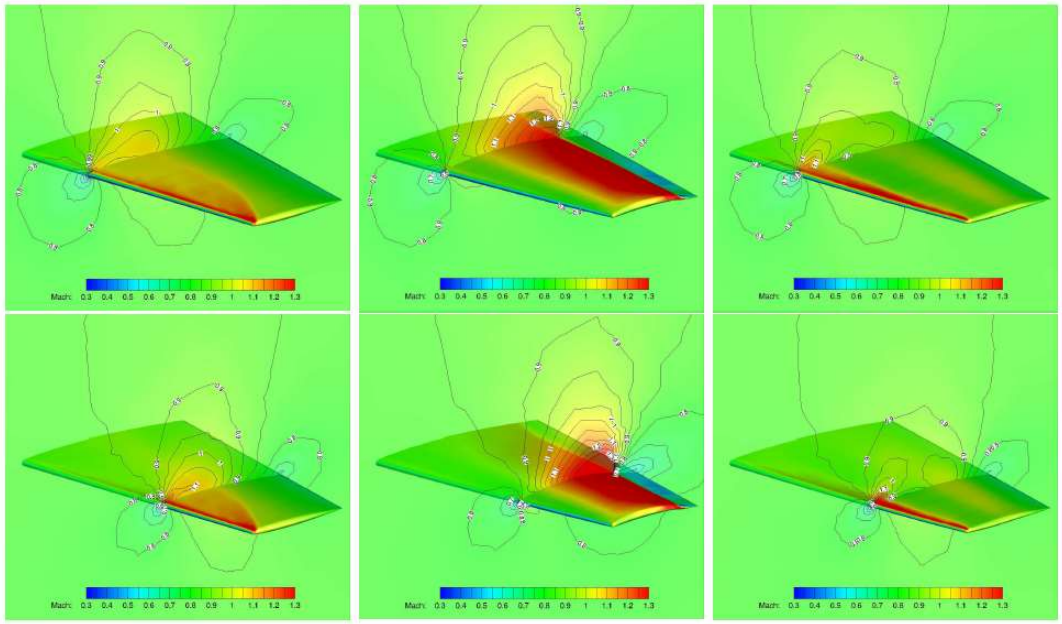


Non-dominated solutions.

Comparisons on Demo Case 2



T^{MM} =30 evaluations on the PSM, before starting the LCPE phase.



Optimized geometries resulted from the DMAEA run. Mach number fields on the surface of the baseline (left), the max. lift (middle) and min. drag (right) wings.

(MA)EAs & the Curse of Dimensionality

Why EAs and MAEAs become very costly in problems with N>>?

- EAs: The evolution slows down in the presence of many DoFs (N>>).
- MAEAs: Difficult to build dependable metamodels or need for many training patterns; consequently:
  - The cost for training metamodel(s) increases a lot.
  - The start of the LCPE phase is delayed.
  - The quality of metamodel-based predictions is damaged.

Possible Remedies:

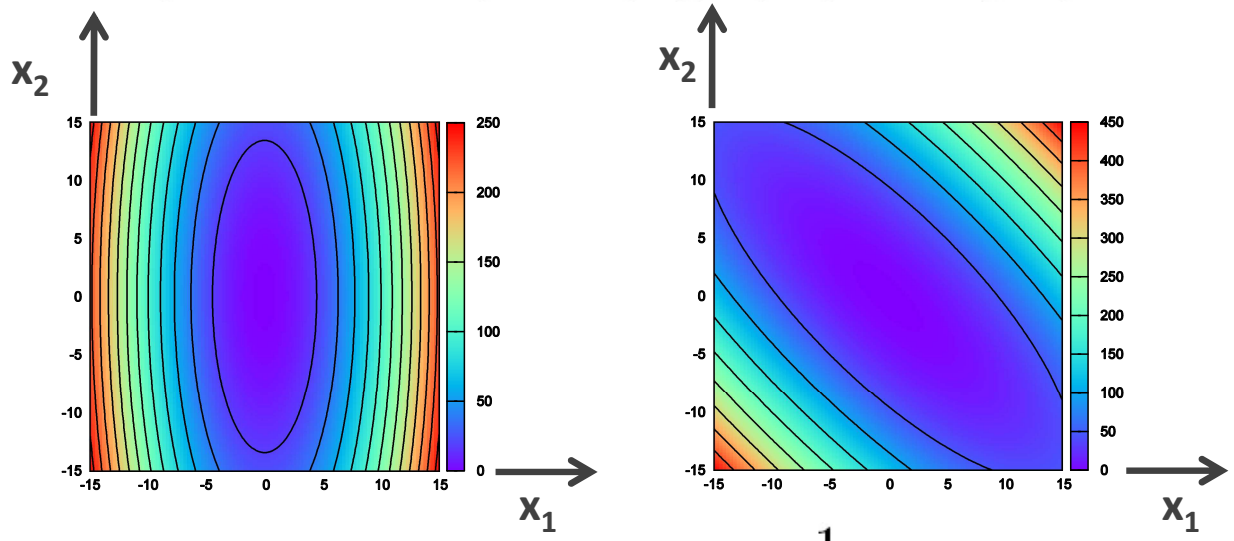
- Artificially transform the problem into a “more separable one”.
- Identify and exclude the less-important DoFs from the evolution .
- Identify and exclude the less-important DoFs from the metamodel training process.

Engineering Optimization, 46:895-911, 2014.

(MA)EAs & the Curse of Dimensionality

Separable vs. Non-Separable Problems:

$$G(x_1, x_2, \ldots, x_N) = g_1(x_1) g_2(x_2) \ldots g_N(x_N)$$



$$F(\vec{x}) = x_1^2 + \frac{1}{9} x_2^2$$

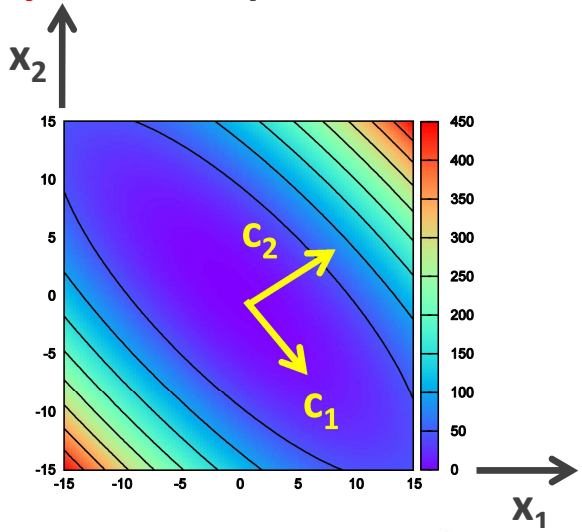
$$F(\vec{x}) = \left[x_1 cos(-\frac{\pi}{4}) - x_2 sin(-\frac{\pi}{4}) \right]^2 + \frac{1}{9} \left[x_1 sin(-\frac{\pi}{4}) + x_2 cos(-\frac{\pi}{4}) \right]^2$$

(MA)EAs & the Curse of Dimensionality

Idea: How to efficiently solve a Non-Separable problem:

Perform a rotation (how??) and make the EA solve for (c_1,c_2), instead of (x_1,x_2).

The Principal Component Analysis can help us!



$$F(\vec{x}) = \left[x_1 cos(-\frac{\pi}{4}) - x_2 sin(-\frac{\pi}{4})\right]^2 + \frac{1}{9}\left[x_1 sin(-\frac{\pi}{4}) + x_2 cos(-\frac{\pi}{4})\right]^2$$

(MA)EAs & the Curse of Dimensionality

Linear Principal Component Analysis (LPCA):

- Let X be a set of M observations $\vec{x} \in \mathbb{R}^N$ of possibly correlated variables. These could be the λ (M=λ) members of the current offspring population.
- Make them have zero mean and unit standard deviation:

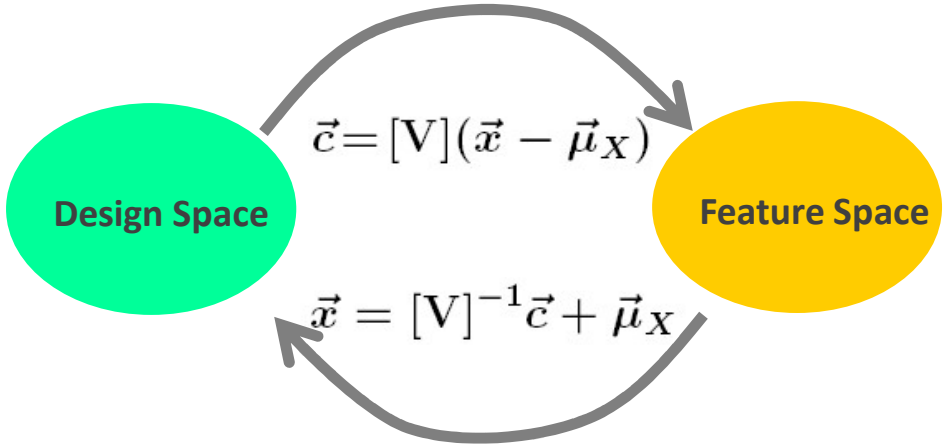
$$E[[\mathrm{X}]] = 0 \qquad E[[\mathrm{X}]^2] = 1$$

- Compute the covariance matrix:

$$[\mathrm{P}]_{N\times N} = [\mathrm{P}] = \frac{1}{M}[\mathrm{X}][\mathrm{X}]^T$$

- Eigen-Decomposition:

$$[\mathrm{P}] = [\mathrm{V}][\Lambda][\mathrm{V}]^T$$

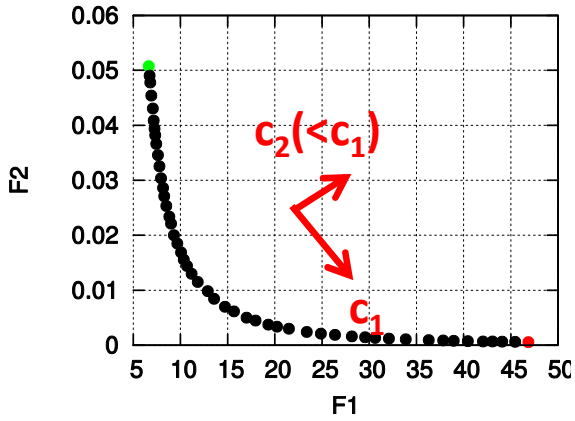


where the eigenvectors are the principal components defining the feature space and the eigenvalues are their variances. The same can be done with Kernel PCA (KPCA-omitted here).

(MA)EAs & the Curse of Dimensionality

EAs and MAEAs with PCA-Driven Evolution Operators

- The current offspring population is used as the set of M=λ observations for the PCA.
- Why? Other options?
- The current parent population is transformed into the feature space (LPCA: M=N or KPCA: M=λ).
- Crossover and mutation apply in the feature space by using different mutation probability per principal component, with increased probability along directions with small variances.
- Transform offspring back to the design space.



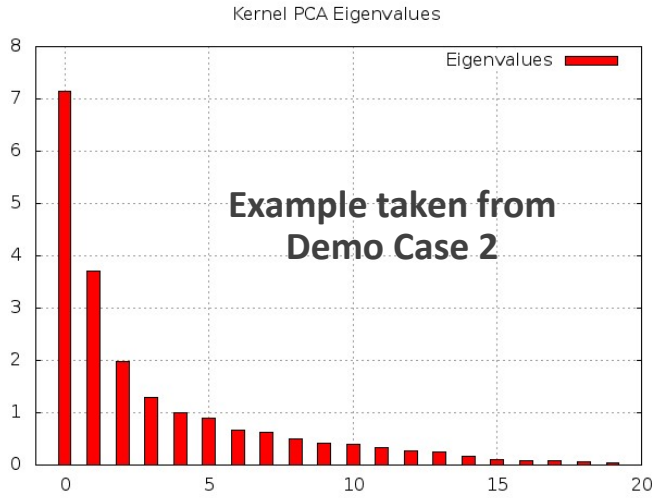
EA(*) EA with evolution operators driven by *PCA
MAEA(*) MAEA with evolution operators driven by *PCA
where *=L (Linear) or *=K (Kernel)

Engineering Optimization, 46:895-911, 2014.

(MA)EAs & the Curse of Dimensionality

MAEAs with PCA-Truncated Metamodels

- **Identification of the most important directions in the feature space, by processing the computed eigenvalues.**
- **Truncation: Keep only the most important components in the feature space and train the metamodels only on them.**

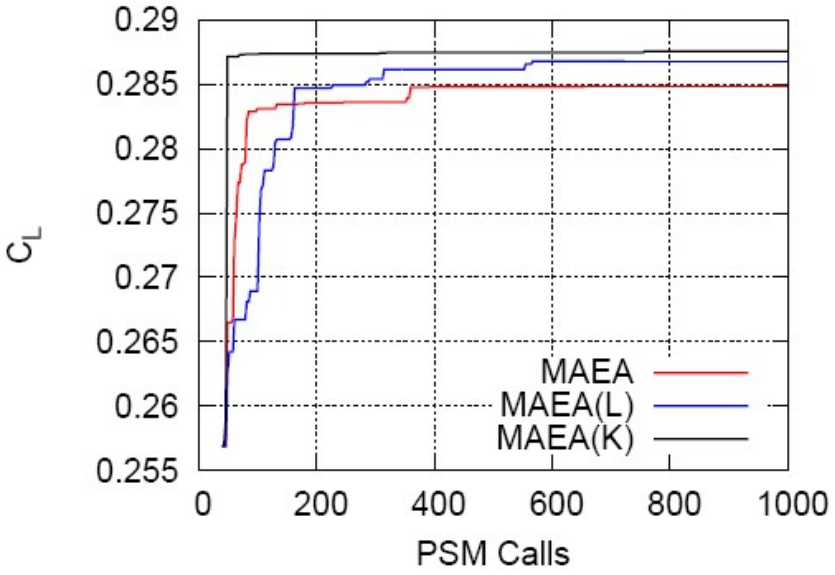


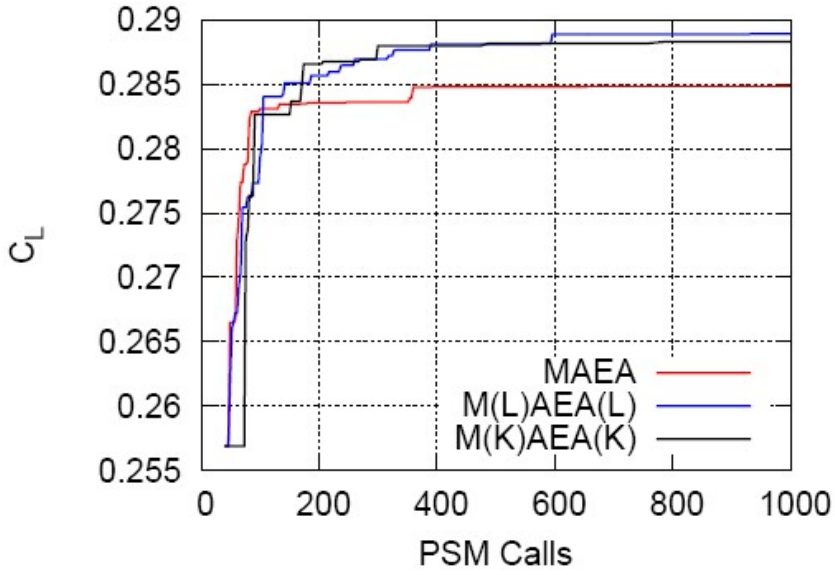
M(*)AEA — **MAEA with truncated metamodels using *PCA where *=L (Linear) or *=K (Kernel)**

Or, in combination:

M(*)AEA(*)

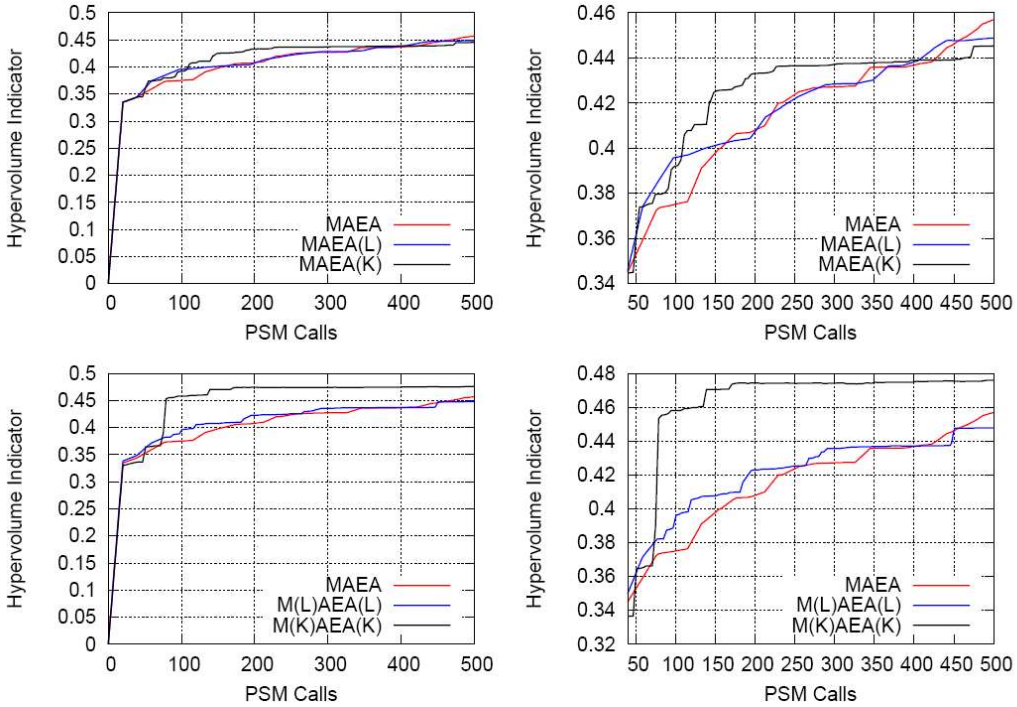
Demo Case 1 Revisited using PCA-based EAs & MAEAs





(20,40) MAEA(L) & (K)
(20,40) M(L)AEA(L), T^{MM}=40, λ_e=3, RBF with 6 inputs
(20,40) M(K)AEA(K), T^{MM}=40, λ_e=3, RBF with 6 inputs

Demo Case 2 Revisited using PCA-based EAs & MAEAs



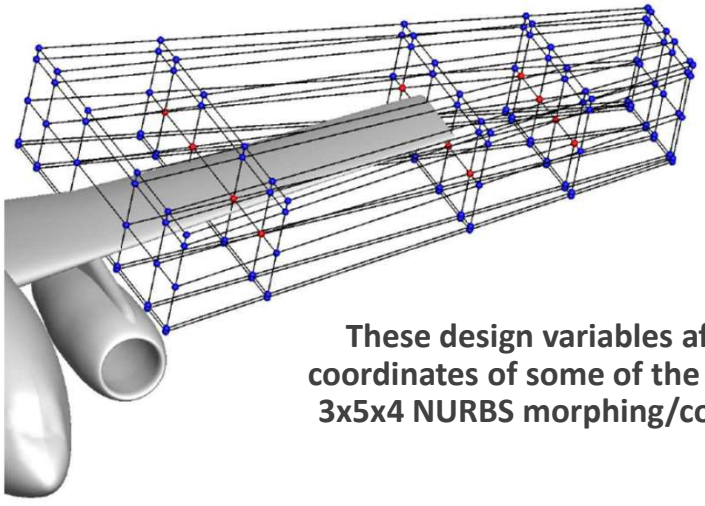
(20,40) MAEA(L) & (K)

(20,40) M(L)AEA(L), T^{MM}=30, λ_e=3, RBF with 12 inputs

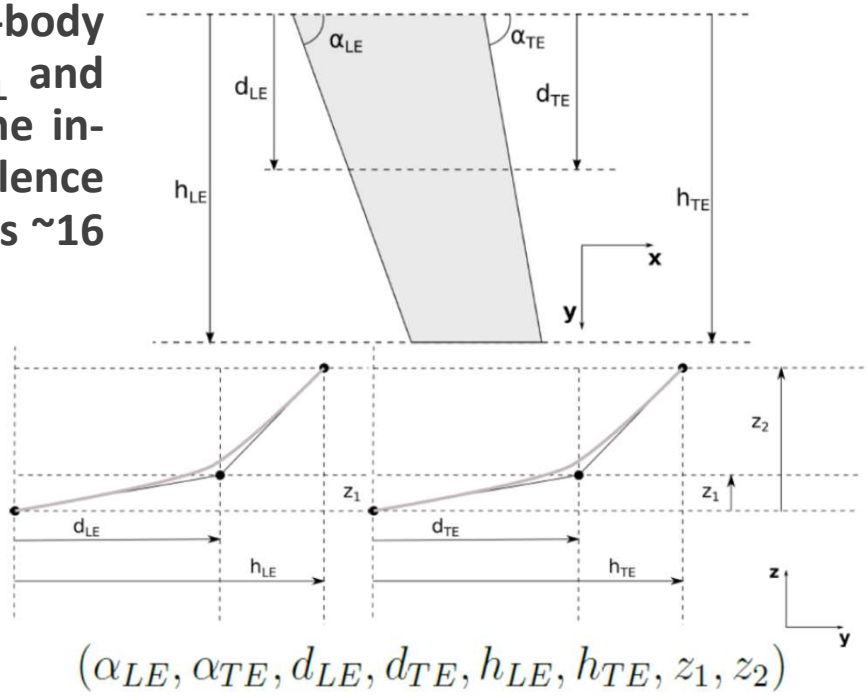
(20,40) M(K)AEA(K), T^{MM}=30, λ_e=3, RBF with 12 inputs

Optimization of an Aircraft Wing-Body Configuration

Re-design of the wing of an aircraft wing-body configuration, with 8 design variables, for max. C_L and min. C_D. $Re_c=10^6$, $M_{inf}=0.75$, $a_{inf}=0^o$. RANS solver (the in-house PUMA code) with the Spalart-Allmaras turbulence model. A single call to the PSM (incl. morphing) takes ~16 min. on an NVIDIA K20 GPU.



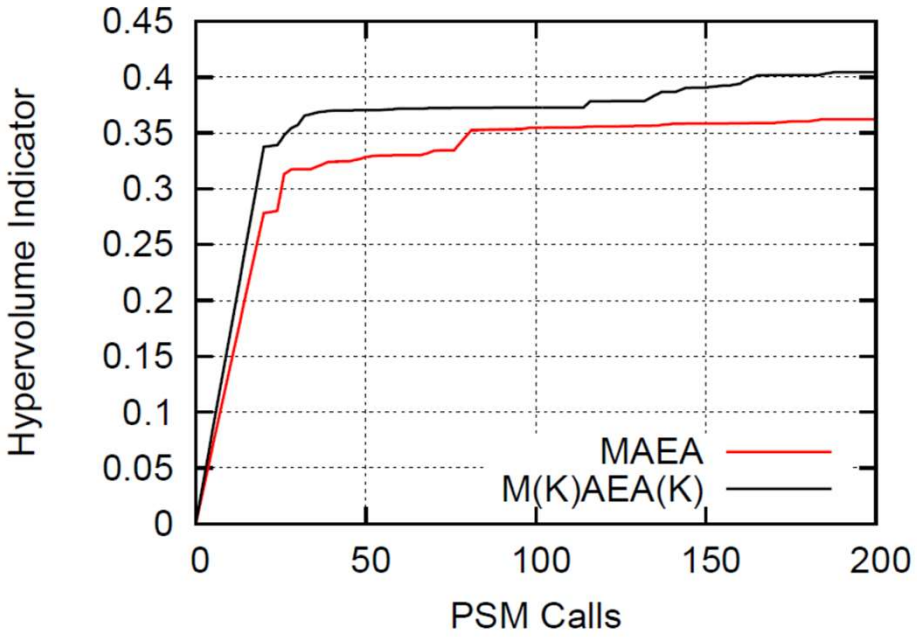
These design variables affect the coordinates of some of the nodes of a 3x5x4 NURBS morphing/control grid

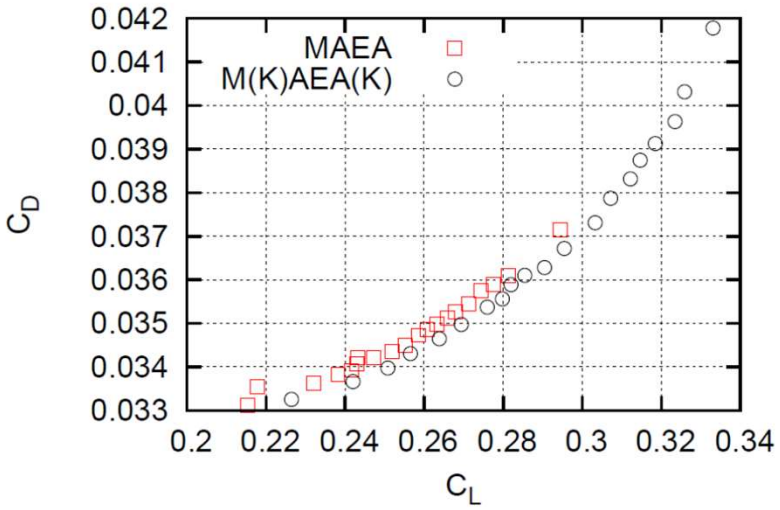


$$(\alpha_{LE}, \alpha_{TE}, d_{LE}, d_{TE}, h_{LE}, h_{TE}, z_1, z_2)$$

Customized parametrization: design variables related to the wing planform (top) & the wing dihedral angles at the leading & training edges (bottom).

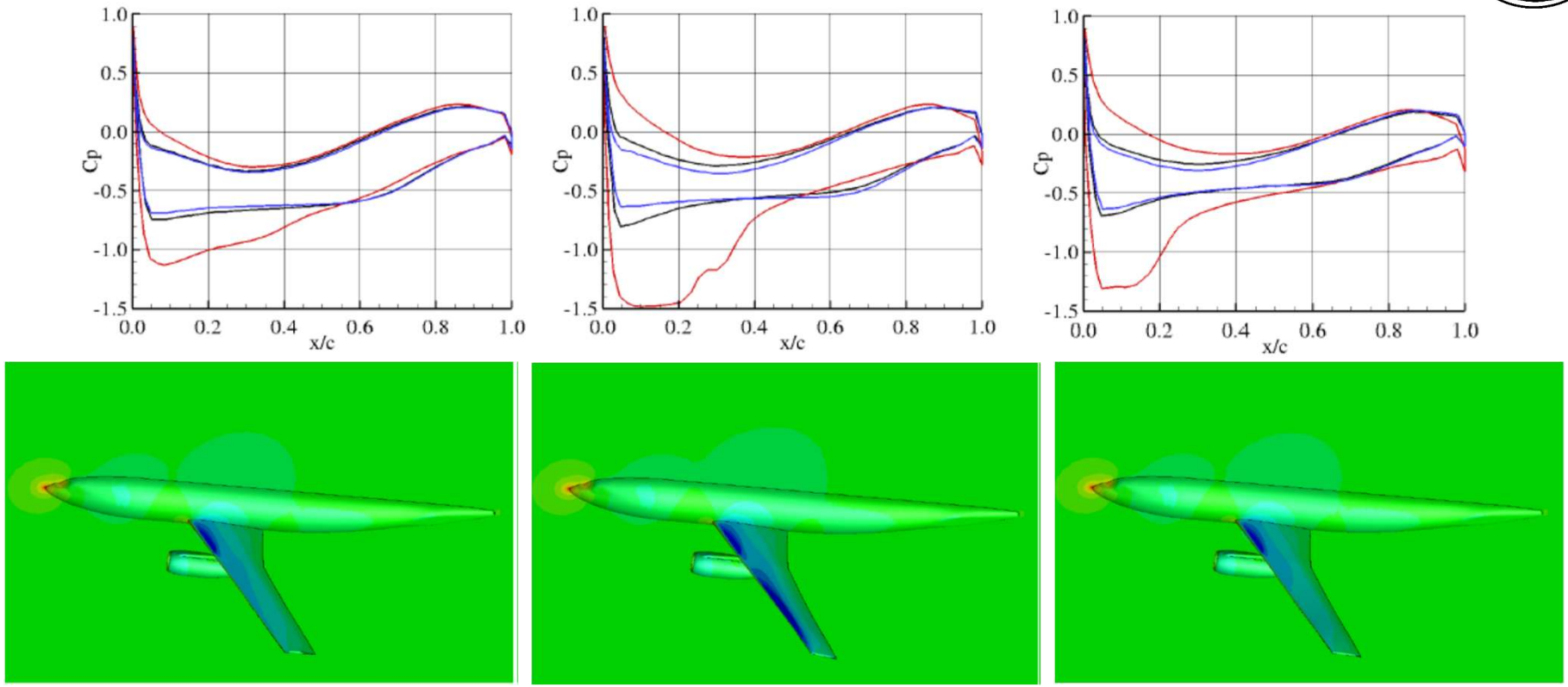
Optimization of an Aircraft Wing-Body Configuration





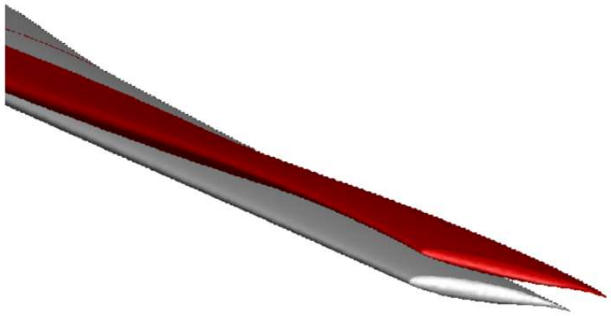
Comparison of the averaged convergence histories for three RNG seeds of the (5,10) MAEA & M(K)AEA(K), in terms of the number of CFD evaluations (PSM Calls). The LCPE phase starts after the first T^{MM}=20 calls to the PSM and the λ_e=4 most "promising" individuals are re-evaluated in each generation. Stopping criterion = 200 CFD evaluations.

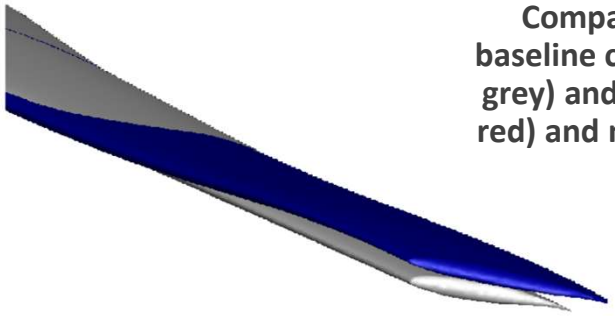
Optimization of an Aircraft Wing-Body Configuration



Comparison of the pressure coefficient distributions & pressure distributions for different wing shapes (a) baseline configuration, (b) max. C_L configuration and (c) min. C_D configuration.

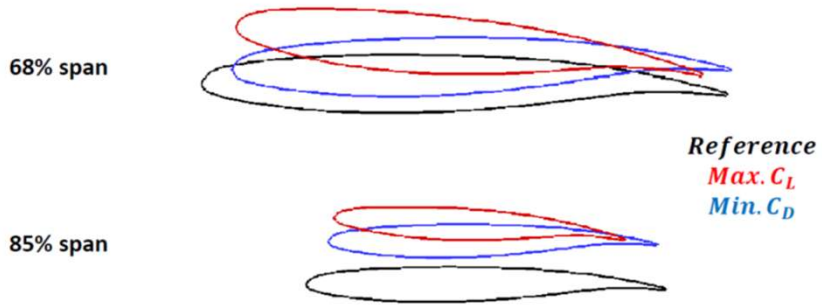
Optimization of an Aircraft Wing-Body Configuration





Comparison of the baseline configuration (in grey) and the max. C_L (in red) and min. C_D (in blue) ones.

Airfoil shapes at three spanwise positions of the baseline configuration (in grey) and the max. C_L (in red) and min. C_D (in blue) ones.

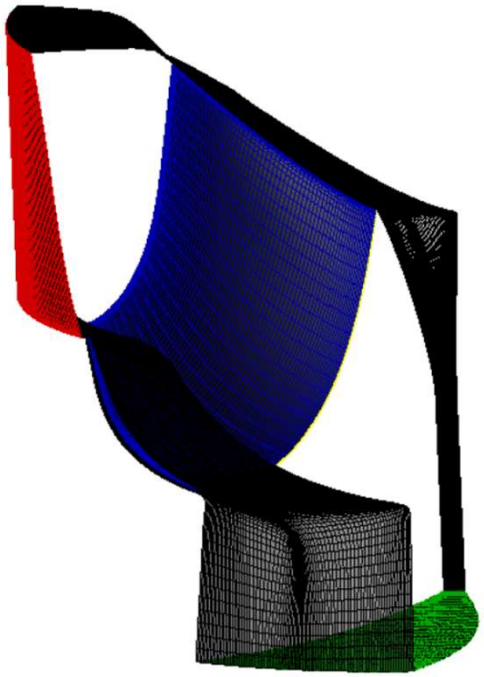


Shape Optimization of a Francis Runner

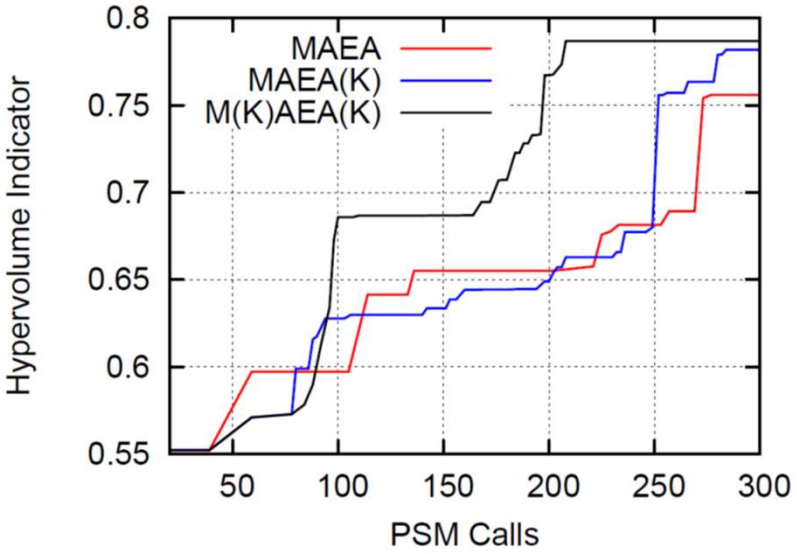
Shape optimization of a Francis runner with two objectives: (a) max. efficiency & (b) min. cavitation (maximize the min. pressure on the blade surface). Inlet flow conditions: V_{inlet}=8.198 m/s, a_{swirl}=22.36^o & a_{axial}=0^o; outlet static pressure 39900 Pa & rotation speed 117.8 rad/s.

The in-house GMTurbo software is used to parameterize the geometry and provide the 75 design variables for the optimization. These correspond to the span-wise distributions of quantities parameterizing the camber surface.

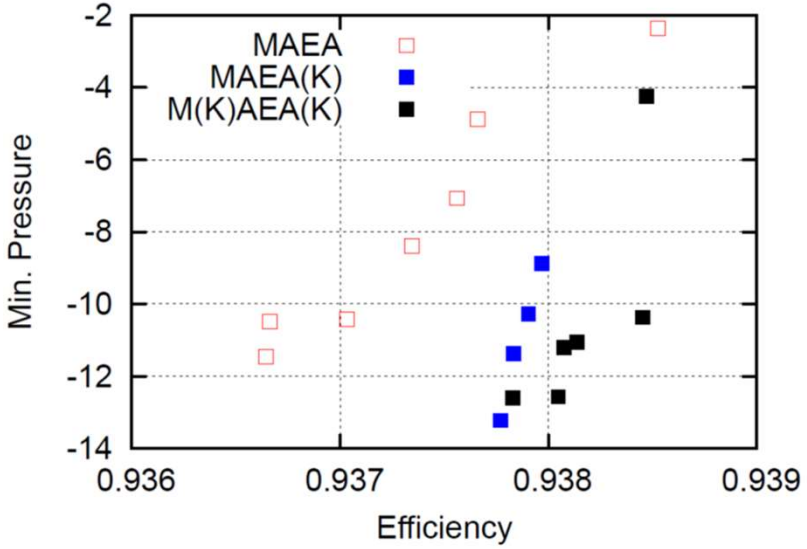
Cost per CFD evaluation: ~1 hr on a single K40 GPU.



Shape Optimization of a Francis Runner

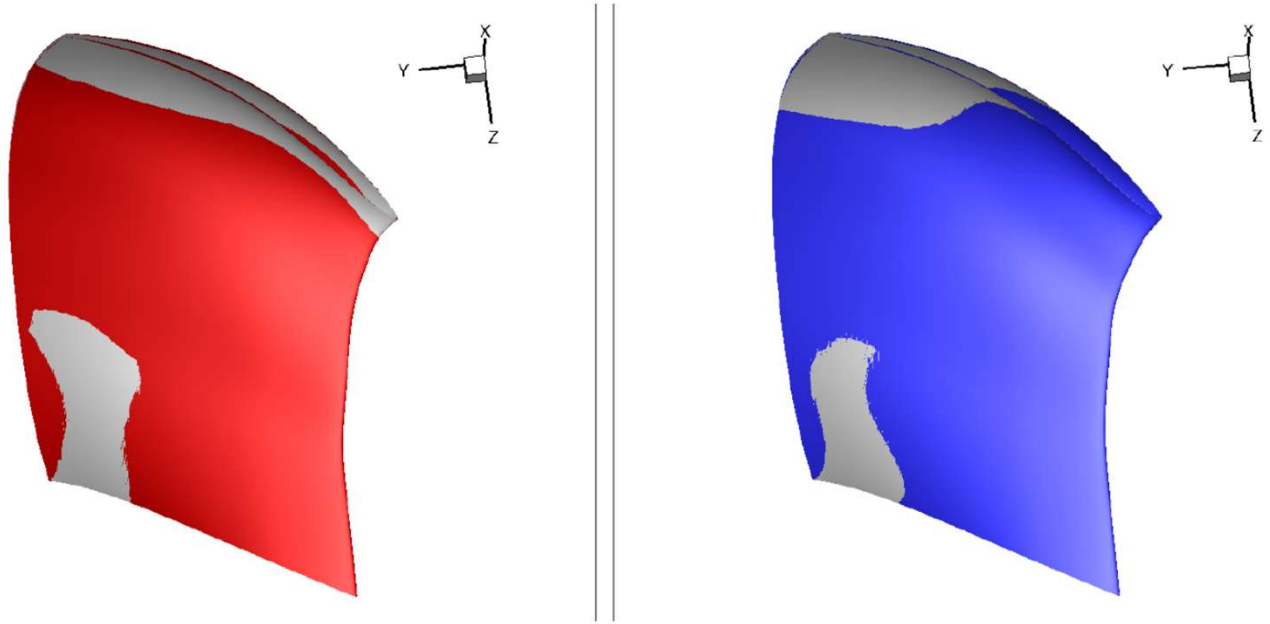


Comparison of the averaged convergence histories for three RNG seeds of the (10,20) EA, MAEA and M(K)AEA(K), in terms of the number of CFD evaluations (PSM Calls). The LCPE phase starts after the first T^{MM}=20 calls to the PSM and the λ_e=2 most "promising" individuals are re-evaluated in each generation. Stopping criterion = 300 CFD evaluations.



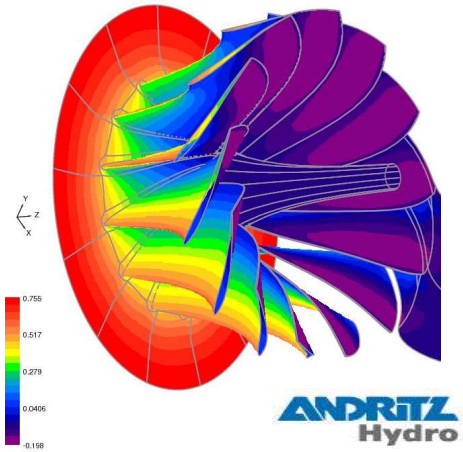
Comparison of the fronts of non-dominated solutions resulted from the (10,20) EA, MAEA and M(K)AEA(K).

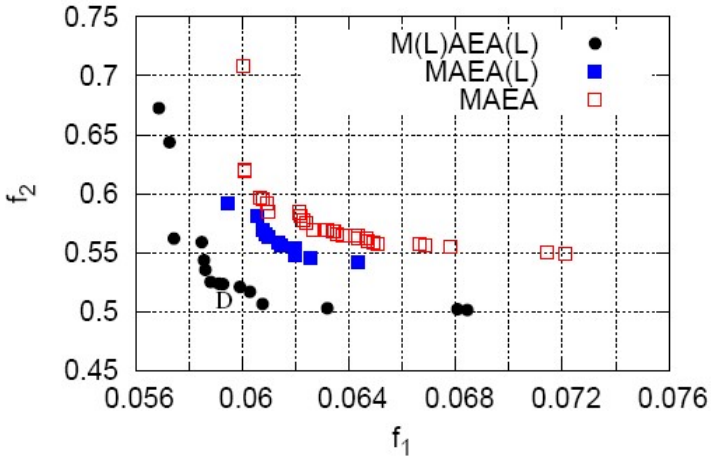
Shape Optimization of a Francis Runner

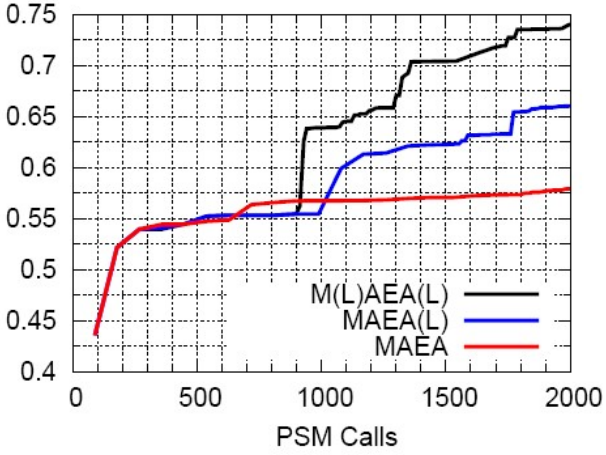


Comparison between the baseline runner (in grey) and those with max. efficiency (in red) and min. cavitation (in blue).

Optimization of a (second) Francis Runner





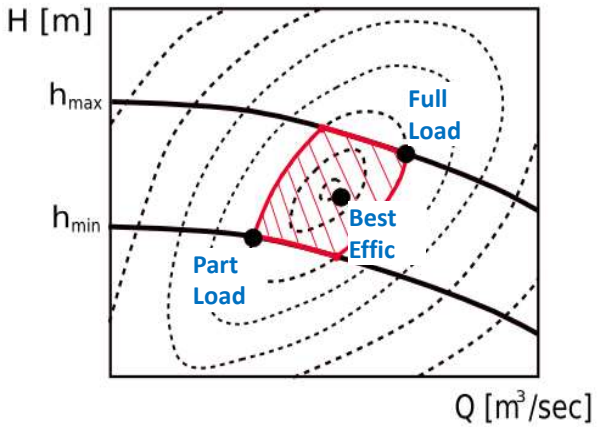


- **Design of the Francis runner, at 3 operating points with two objectives: (a) exit velocity profiles' quality and (b) uniformity of the blade loading. Two constraints (head and cavitation). There are 372 design variables, in total!**
- **Comparison of fronts of non-dominated solutions obtained at the same number of evaluations on the PSM (same CPU cost).**
- **Due to the extremely high problem dimension, the use of M(L)AEA(L) becomes absolutely necessary!**

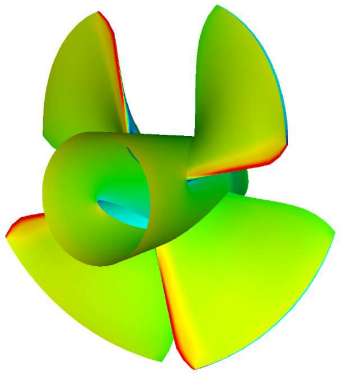
Design of HYDROMATRIX ®

- **HYDROMATRIX®: a number of “small”, axial flow turbine generator units comprising a factory assembled grid or “matrix”.**
- **Advantages compared to conventional designs (lower cost to power ratio):**
- **Minimization of the required civil construction works.**
- **Minimum time for project schedules, construction and installation.**
- **Small geological and hydrological risks. Minimum environmental inflict .**

Design of HYDROMATRIX ®

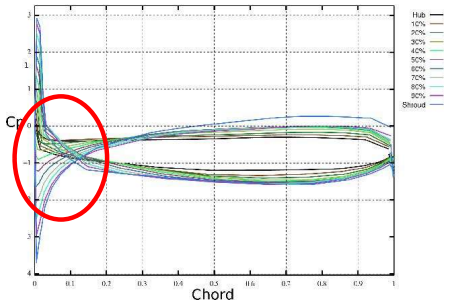


Weights			
	peak point (OP_1)	part load (OP_2)	full load (OP_3)
$w_1^{OP_i}$	1.0	0.0	0.0
$w_2^{OP_i}$	1.0	0.0	0.0
$w_3^{OP_i}$	0.2	0.0	0.0
$w_4^{OP_i}$	1.0	1.0	1.0
$w_5^{OP_i}$	0.0	100.0	100.0
W_{OP_i}	1.0	0.1	0.1

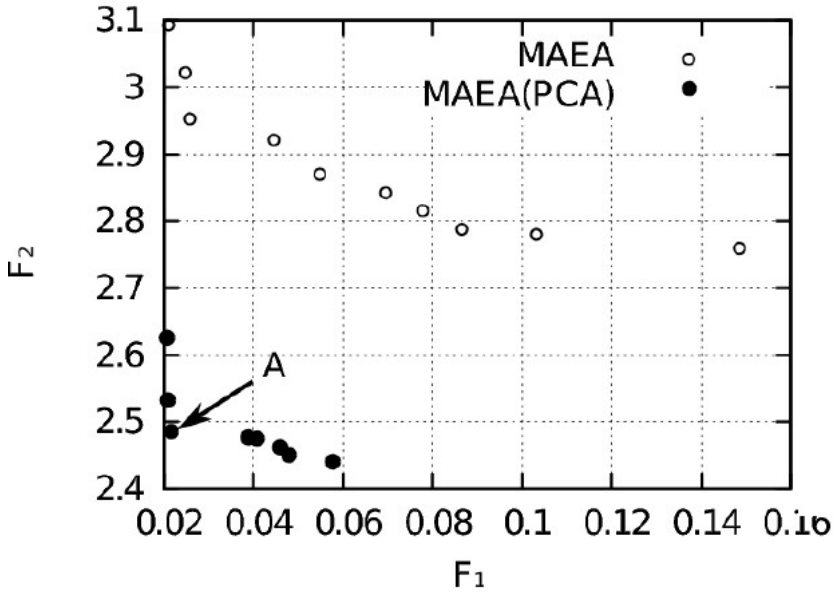


- **<u>52 design variables</u>**
- **Five quantities of interest (metrics): (f_1,f_2) : Given swirl and axial velocity distributions at the exit. (f_3): Uniform loading. (f_4): cavitation index. (f_5): pumping area.**
- **(5 metrics) x (3 operating points) = 15 objectives, cast as a two-objective problem.**
- **min. F_1(f_1,f_2, at the 3 OPs) , min. F_2(f_3,f_4,f_5, at the 3 OPs)**

Design of HYDROMATRIX ®



Pumping area.



Computations with the same budget.

List of Communicated Messages

- **EAs or any kind of stochastic population-based metaheuristics (Genetic Algorithms, Evolution Strategies, Particle Swarm Optimization, Differential Evolution, etc. etc) may locate global optimal solutions at a high cost as a great number of evaluations on the PSM (CFD, CSM, CAA, etc code) are necessary.**
- **Assisting the above metaheuristics with surrogate models is a "must" for reducing the optimization turn-around time. The type of surrogate models (metamodels: polynomial regression, neural networks, Gaussian processes, etc etc) and, in particular, the way these are implemented within the search algorithm are critical. Cost reduction by even an order of magnitude can be achieved (from EAs to MAEAs).**
- **A well-coordinated distributed search is also beneficial (from MAEAs to DMAEAs).**
- **As evolution slows down in the presence of a great number of design variables (curse of dimensionality), smart use of Principal Component Analysis (PCA), either for allowing the application of evolution operators to a "transformed design space" or for reducing the inputs seen by the metamodels may help a lot. Linear of kernel PCA can be used (from DMAEAs to PCA-Assisted DMAEAs).**

List of Communicated Messages

- **Should a gradient-based optimization (such an adjoint method in CFD) be available, hybridization may help. In MOO problems, computing the descent direction that improves the current front of non-dominated solutions is challenging (from PCA-Assisted DMAEAs to Hybrid PCA-Assisted DMAEAs).**
- **Other hybrid schemes, such as multi-level or hierarchical search, can be used; not shown here (find more in *http://147.102.55.162/research/pubs.html*).**
- **Parallelization and, in particular, asynchronous search (that overcomes the generation synchronization barrier) may help a lot; not shown here.**
- **It is important that all these techniques can be combined into a single stochastic population-based search algorithm and benefits are practically superimposed.**
- **PCOpt/NTUA provides, free of charge, EASY licenses to academic groups or research institutes, upon request.**

Not Included in this Lecture

Asynchronous EAs & MAEAs:

- *Engineering Optimization, 41:241-257, 2009.*
- *Genetic Programming and Evolvable Machines, 10(4):373-389, 2009.*

Metamodel-Assisted Memetic Algorithms (MAMAs):

- *Engineering Optimization, 41(10):909-923, 2009.*

Other Hierarchical Schemes:

- *Engineering Optimization, 41(11):1037-1049, 2009.*

etc.