NATIONAL TECHNICAL UNIVERSITY OF ATHENS

Parallel CFD & Optimization Unit

Laboratory of Thermal Turbomachines

Reduced-Cost Evolutionary Algorithms For Industrial Applications

ΜΕΘΟΔΟΙ ΑΙΤΙΟΚΡΑΤΙΚΗΣ ΚΑΙ ΣΤΟΧΑΣΤΙΚΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ & ΕΦΑΡΜΟΓΕΣ

Kyriakos C. GIANNAKOGLOU, Professor NTUA

kgianna@central.ntua.gr

http://velos0.ltt.mech.ntua.gr/research/

V. ASOUTI

ΕΙΣΑΓΩΓΗ

Η παρουσίαση αυτή καλύπτει την ύλη που διδάσκεται σε ένα τρίωρο του Μεταπτυχιακού Μαθήματος «Μέθοδοι Αιτιοκρατικής και Στοχαστικής Βελτιστοποίησης & Εφαρμογές», στα ΔΠΜΣ «Υπολογιστική Μηχανική» και «Εφαρμοσμένες Μαθηματικές Επιστήμες» του ΕΜΠ.

Στοχεύει να δείξει ότι οι Εξελικτικοί Αλγόριθμοι (ΕΑ) στην κλασική τους μορφή είναι αργές μέθοδοι αλλά υπάρχουν πλέον «έξυπνες» τεχνικές που μπορούν να τους κάνουν ικανούς να παράγουν τη βέλτιστη λύση σε αποδεκτό αριθμό αξιολογήσεων, να χρησιμοποιούν βέλτιστα τα πολυεπεξεργαστικά συστήματα, να συνδυάζονται με άλλες μεθόδους βελτιστοποίησης κλπ.

Μόνο με μεθόδους όπως αυτές που ακολουθούν, οι ΕΑ μπορούν να λύνουν μεγάλης κλίμακας προβλήματα βελτιστοποίησης σε αποδεκτό χρόνο και έτσι να γίνονται αποδεκτοί (δεδομένων όλων των άλλων πλεονεκτημάτων τους) για βιομηχανικούς υπολογισμούς.

ΕΙΣΑΓΩΓΗ

Κρίθηκε σκόπιμο οι διαφάνειες που καλύπτουν την ύλη να παραμείνουν στην Αγγλική γλώσσα.

Δίνονται βιβλιογραφικές αναφορές για παραπάνω εμβάθυνση, ανάλογα με τα ενδιαφέροντα του σπουδαστή.

Οι εφαρμογές είναι σχεδόν αποκλειστικά από την περιοχή της βελτιστοποίησης μέσω Υπολογιστικής Ρευστοδυναμικής (CFD-based Optimization, λόγω προφανώς της ενασχόλησης της Μονάδας μας με αυτό το αντικείμενο. Η γενίκευση όμως των μεθόδων σε άλλες εφαρμογές, κυρίως δε σε αυτές που το λογισμικό αξιολόγησης΄έχει μεγάλο υπολογιστικό κόστος) είναι άμεση.

EAs: Pros & Cons

- ⊕ Ready to accommodate black-box analysis-evaluation software …
- ⊕ Can easily (efficiently??) handle constraints …
- ⊕ Appropriate for computing Pareto fronts in MOO problems
- ⊕ Supports parallel evaluations on multiprocessor systems

- ⊖ Requires a great number of evaluations …

EAs: Low-Budget Optimization

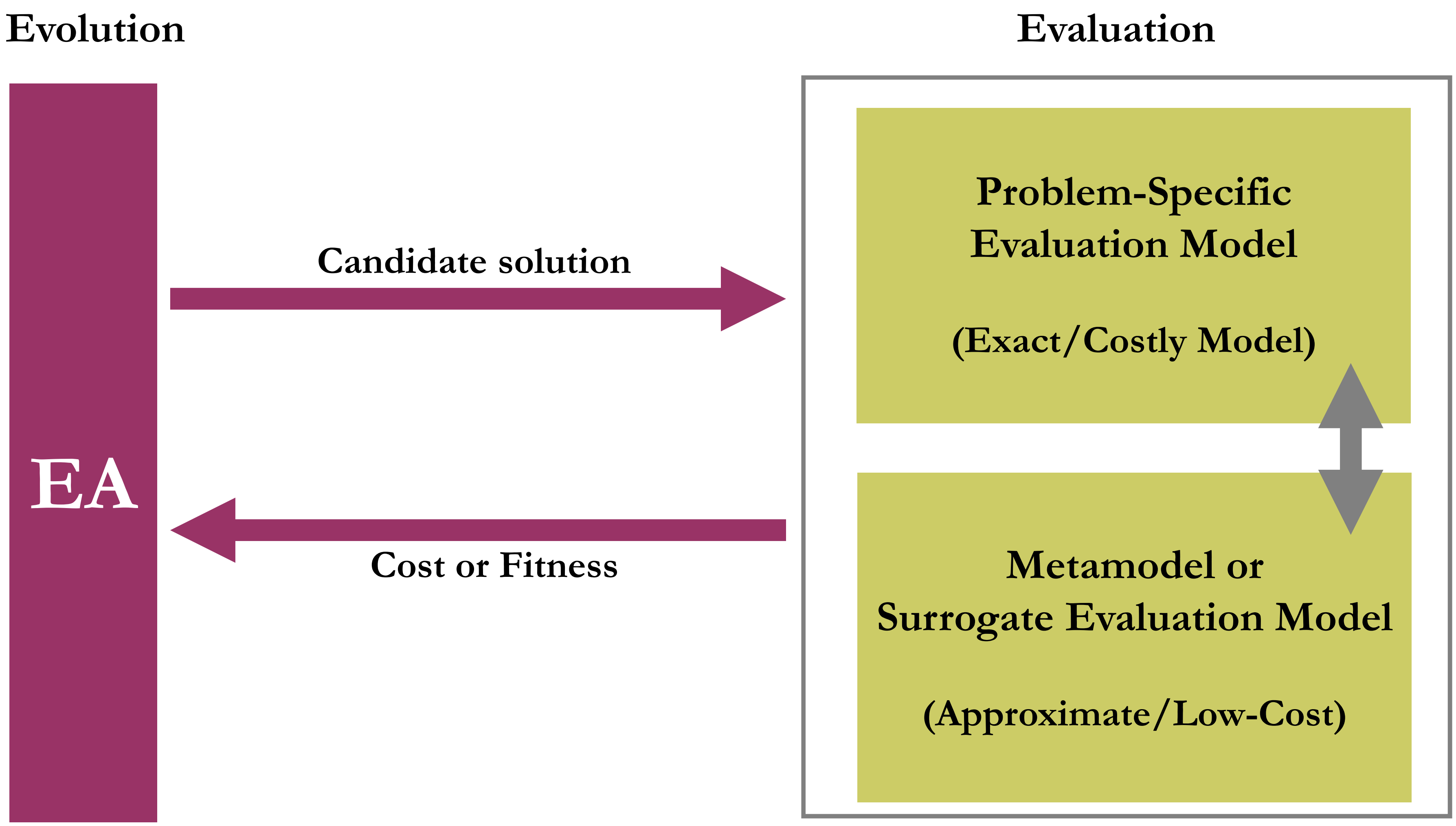
The industrial use of EAs requires enhancement through methods capable of:

- Cutting-down the number of (CFD, etc based) evaluations
- Reducing the wall-clock time of the optimization
- Doing both of them!

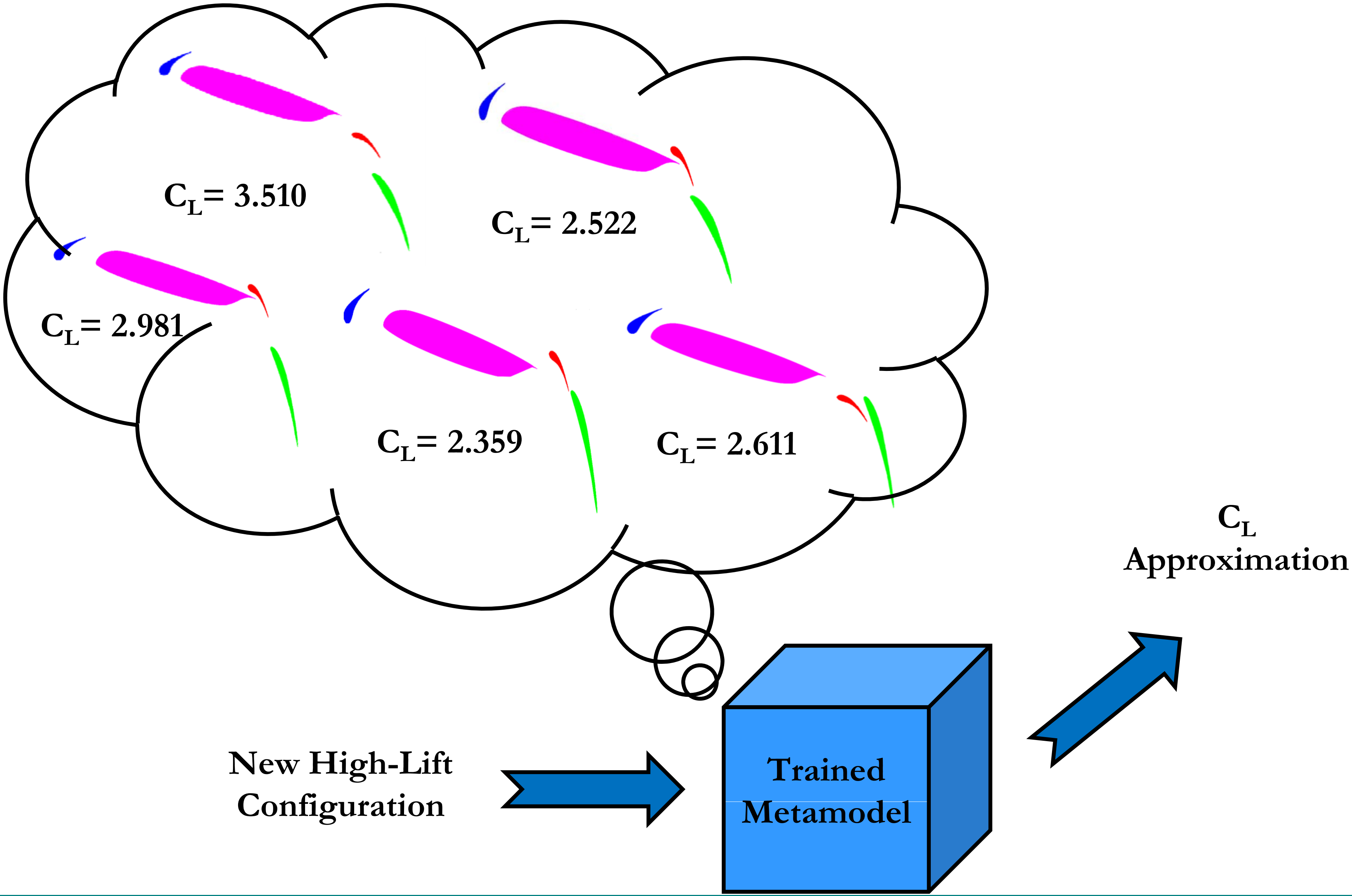
Such as:

- Employing metamodels or surrogate evaluation models
- Working with distributed search methods
- Employing hierarchical search
- Improving the evolution operators
- Exploiting, as much as possible, a multiprocessor platform (heterogeneous?)
- Dimensionality reduction in “metamodels” & “evolution”
- Combining some or all of the above!

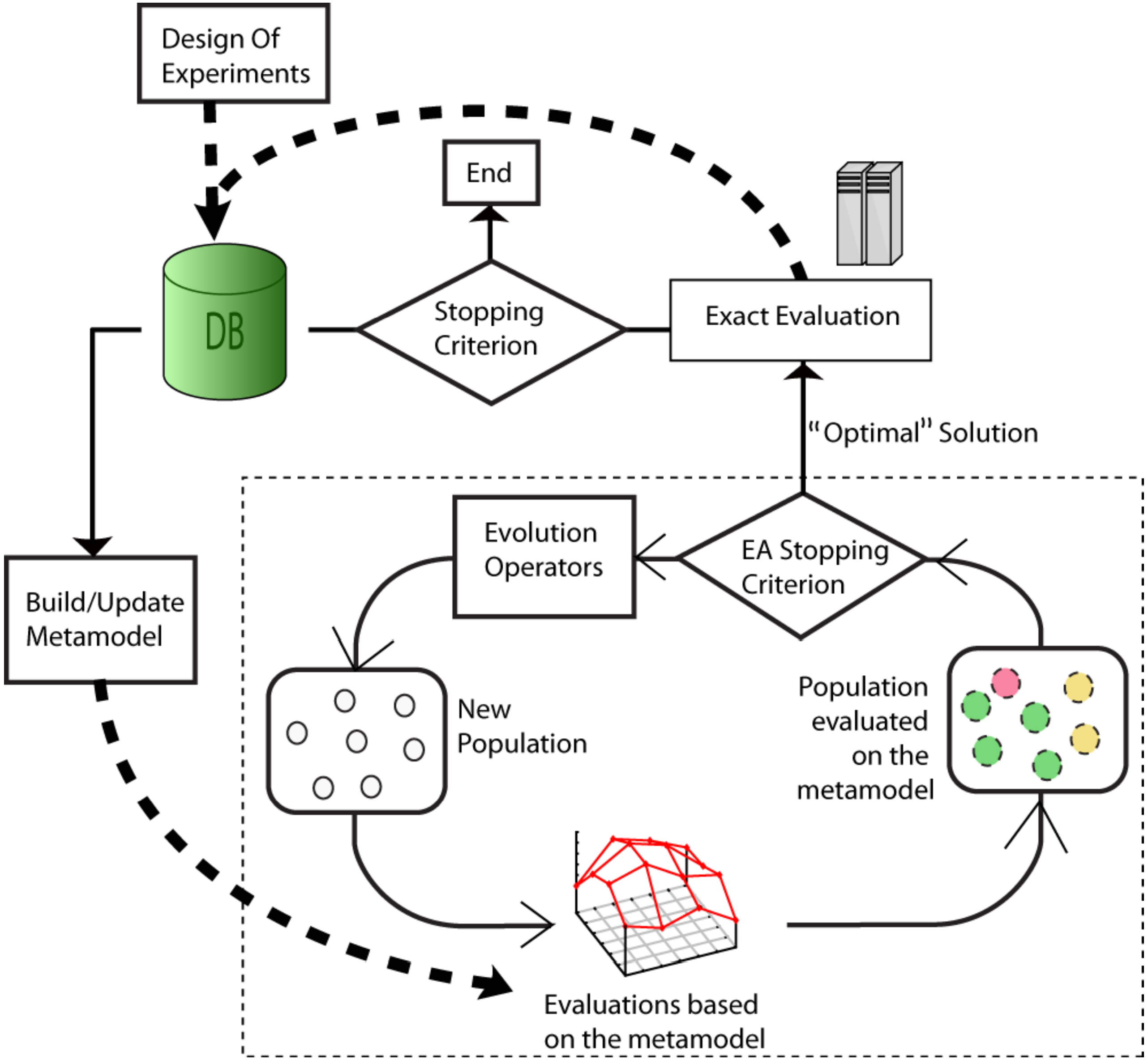
Metamodel-Assisted EAs (MAEAs)



The Role of Metamodels during the Evolution

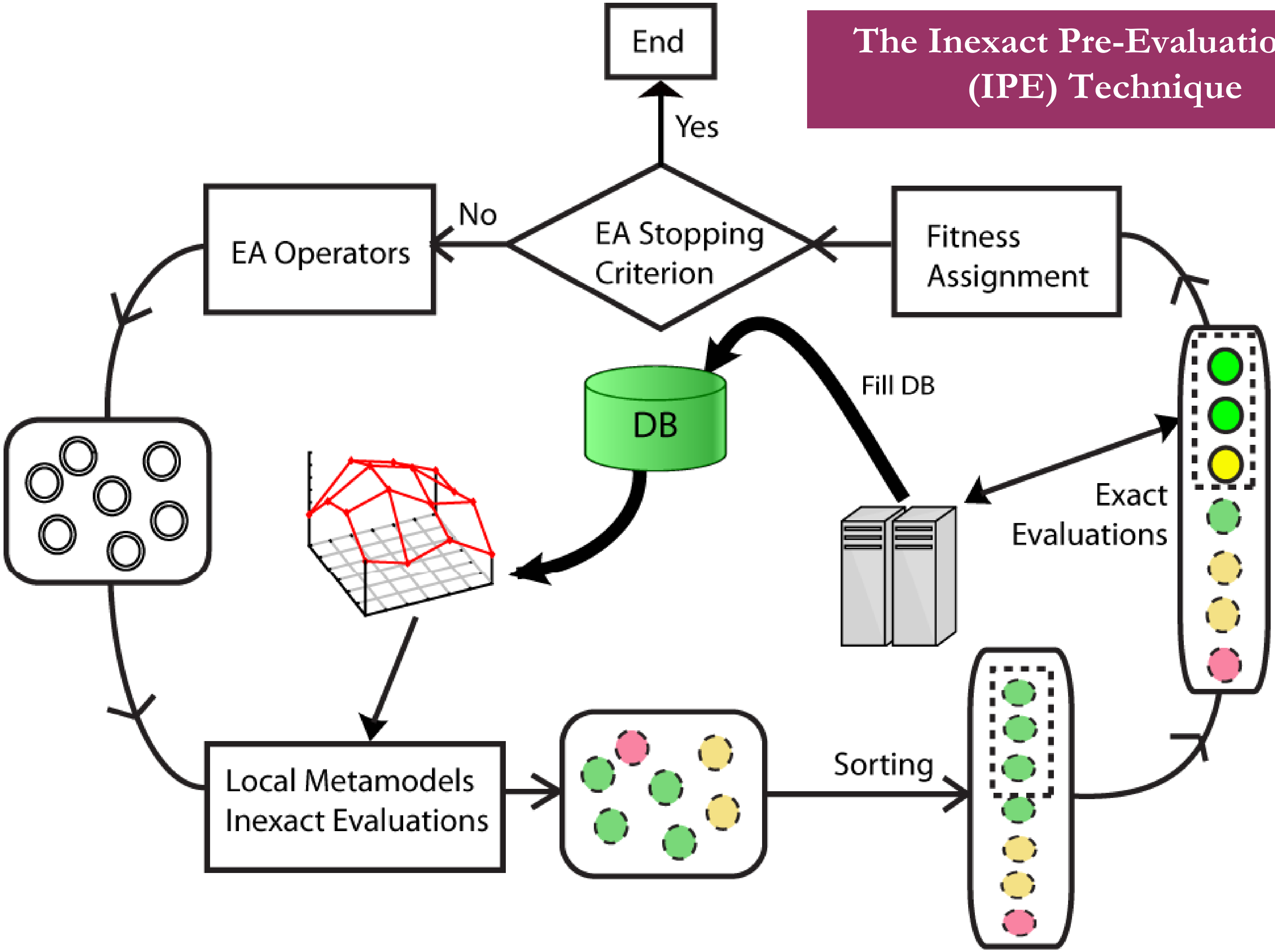


MAEAs based on *Off-Line* Trained Metamodels

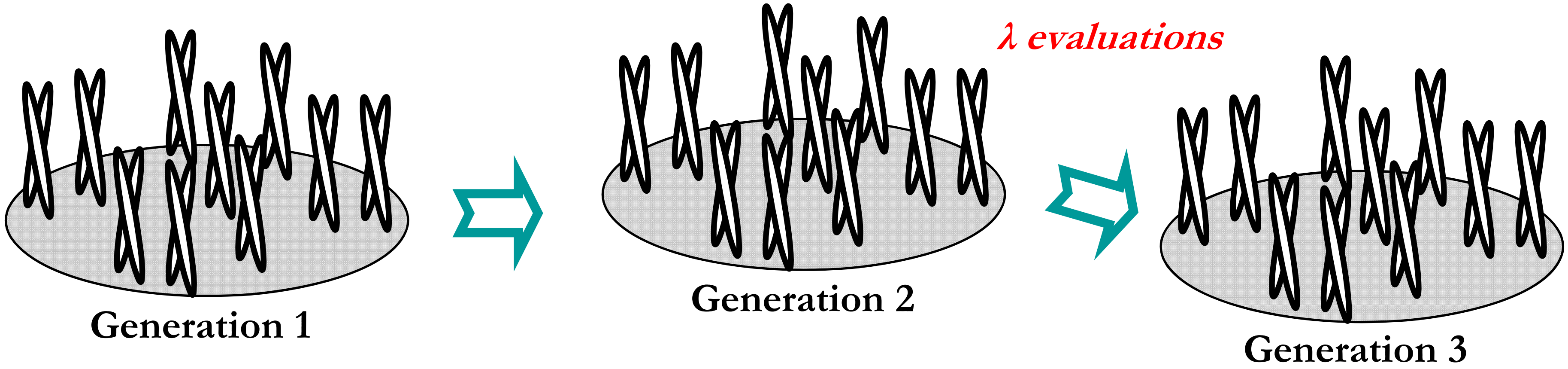


MAEAs based on *On-Line* Trained Metamodels

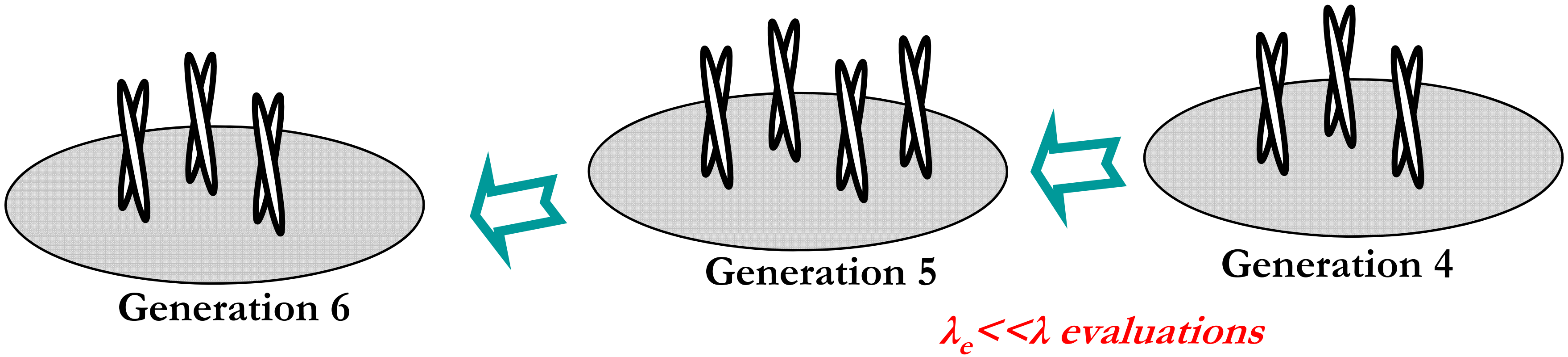
The Inexact Pre-Evaluation (IPE) Technique



MAEAs with Inexact Pre-Evaluation (IPE)



More generations are needed; however, apart from the very first ones, the number of calls to the expensive/ accurate evaluation tool per generation corresponds to a small percentage of the population only!



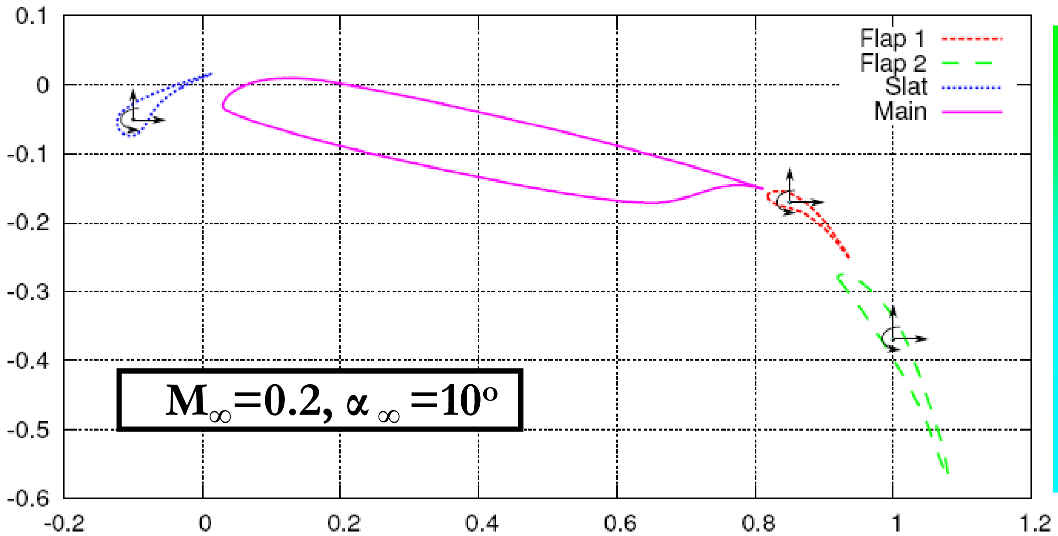
K.C. GIANNAKOGLOU: 'Design of Optimal Aerodynamic Shapes using Stochastic Optimization Methods and Computational Intelligence', Int. Review Journal Progress in Aerospace Sciences, Vol. 38, pp. 43-76, 2002.

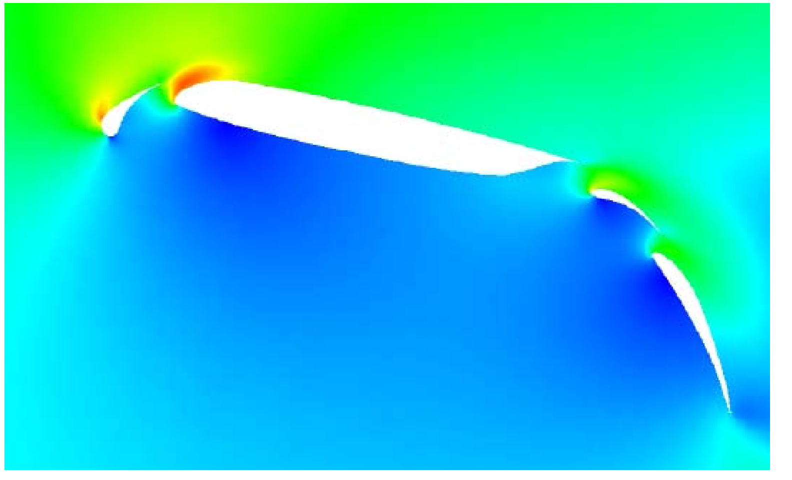
Possible Metamodel Types

- Polynomial Regression
- Artificial Neural Networks
  - Radial Basis Functions
  - Multilayer Perceptrons
- Kriging Model
- Fitness Inheritance
- Support Vector Machines
- …

MAEA based on the IPE technique – An Example

Optimal Deployment of a 4-element airfoil for maximum lift

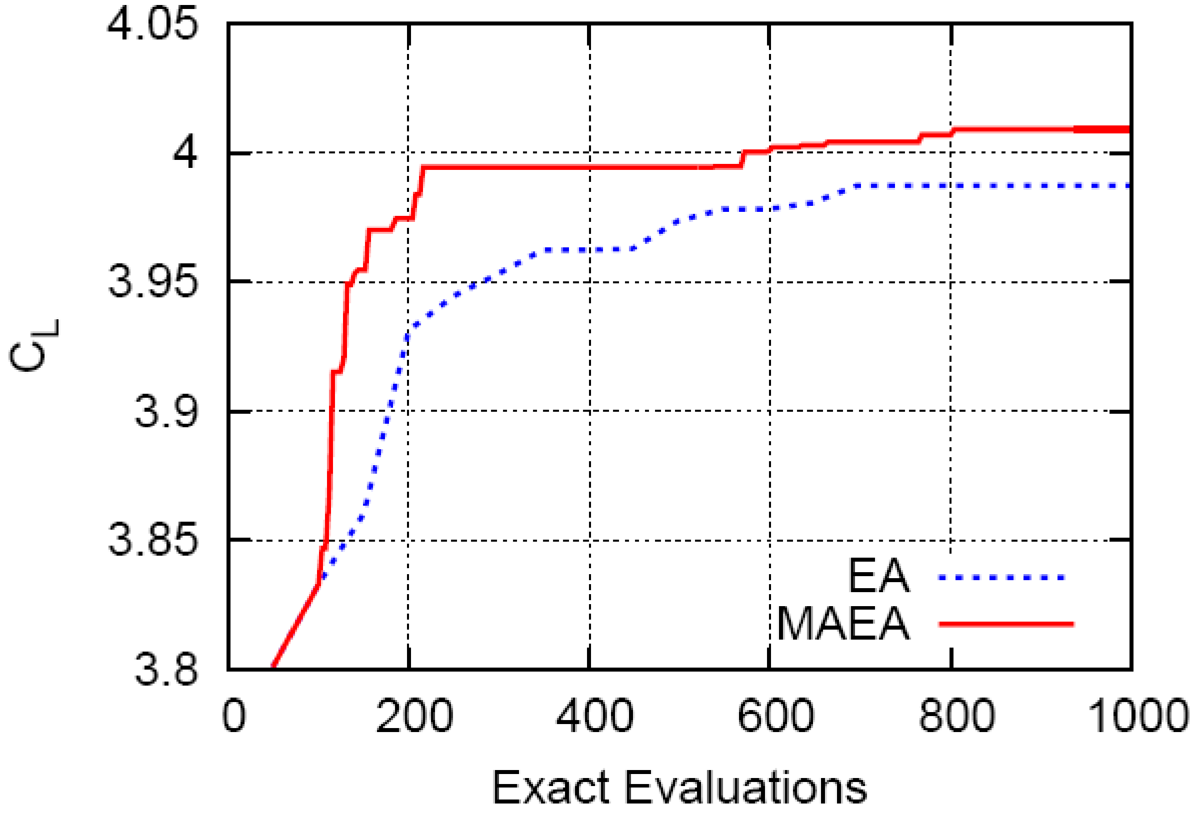




Design variables:
Two displacements and a rotation angle per element,
9 design variables in total.

MAEA based on the IPE technique – An Example

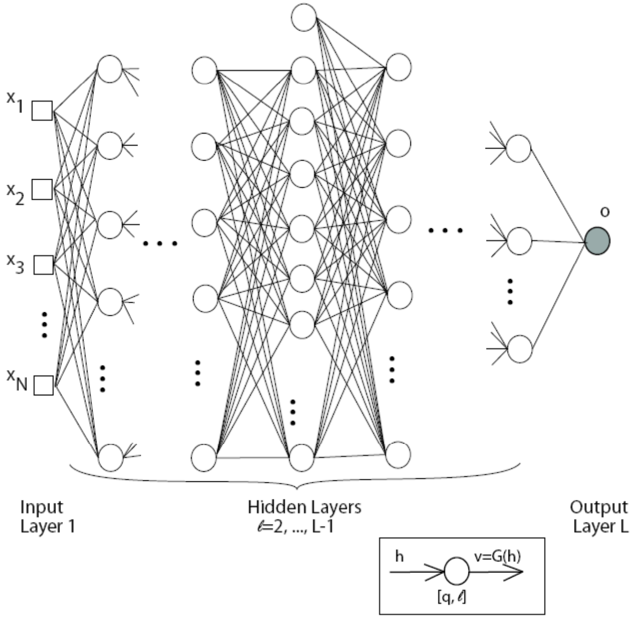
Optimal Deployment of a 4-element airfoil for maximum lift



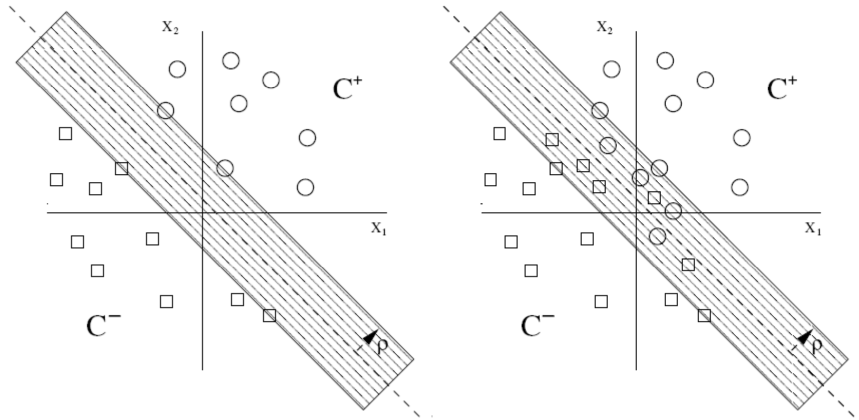
Comparison of (20,50) EA & (20,50) MAEA,
in which λ_e=6 & DB_{start}=100.

Other Metamodels

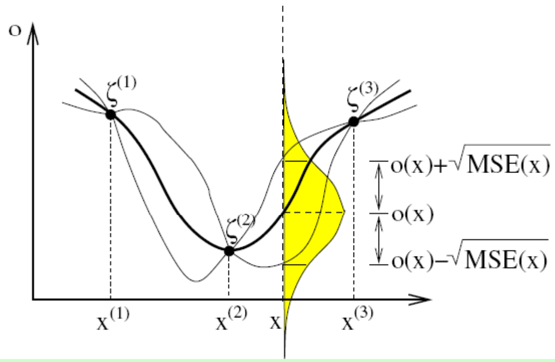
Multilayer Perceptrons



Support Vector Machines (SVM)



Kriging Model

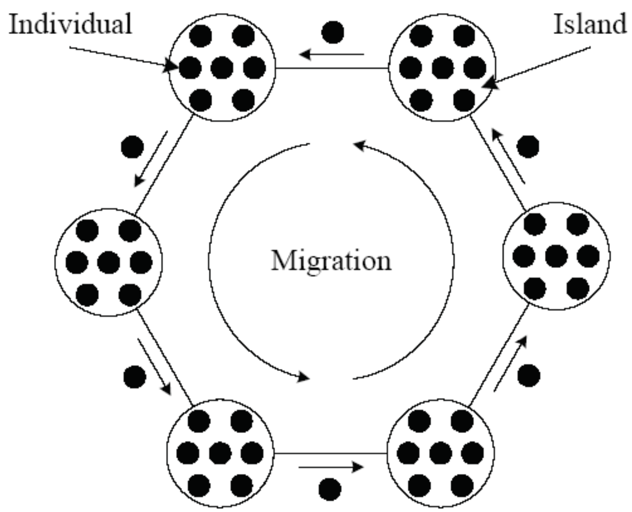


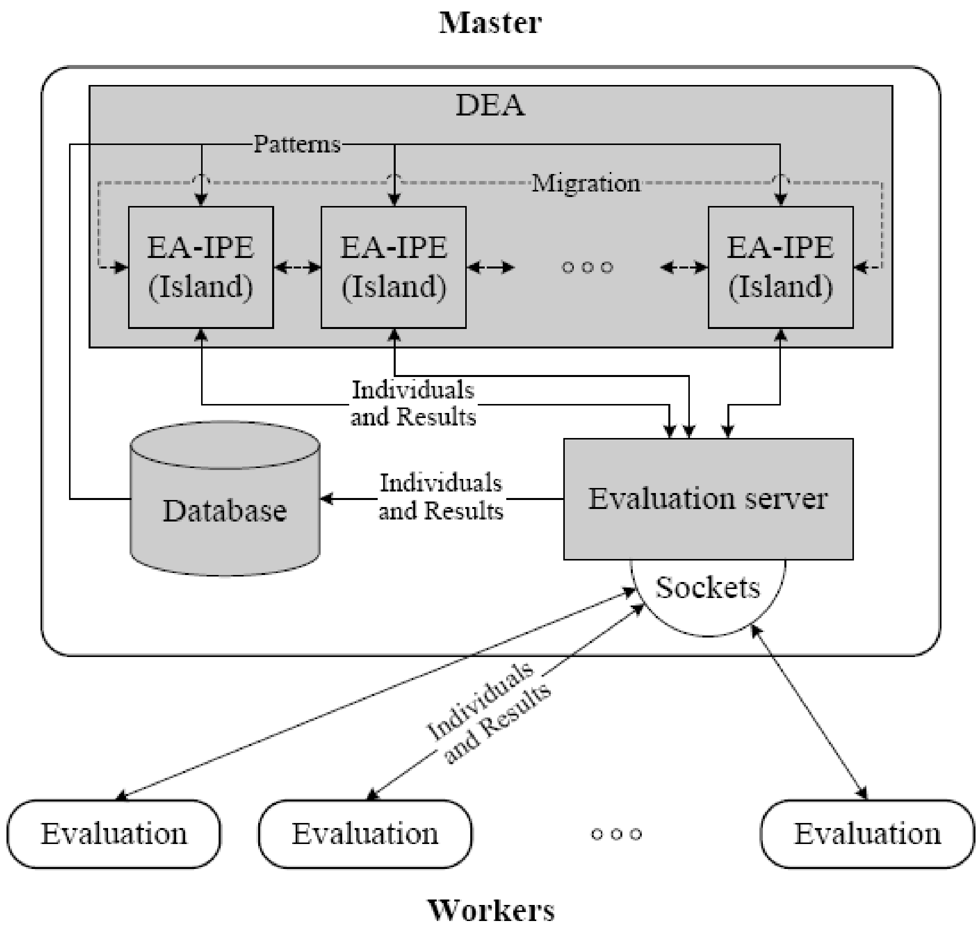
Many interesting IPE variants can be devised using the kriging metamodel.

Not to be discussed further in this lecture!

M. EMMERICH, K.C. GIANNAKOGLOU and B. NAUJOKS: 'Single- and multi-objective evolutionary optimization assisted by Gaussian random field metamodels', IEEE Transactions on Evolutionary Computation, Vol. 10, pp. 421-439, 2006.

Distributed Metamodel-Assisted EAs (DMAEAs)





Basic issues:

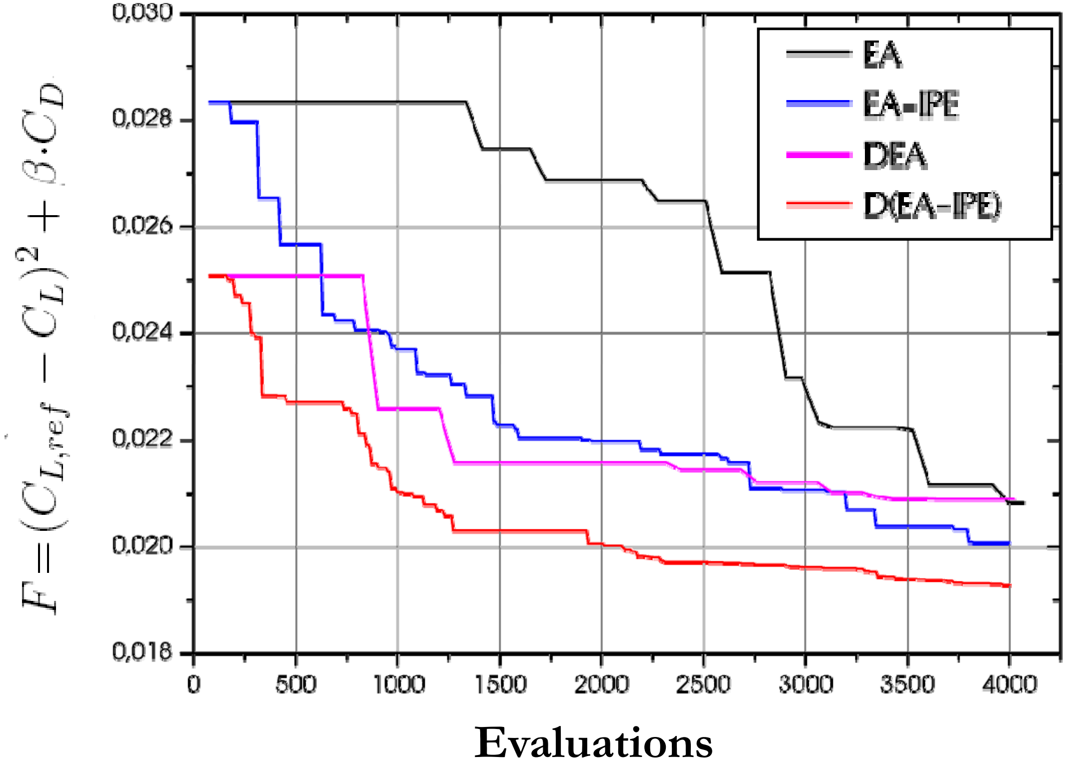
- Number of demes or islands
- Communication topology
- Communication frequency
- Migration algorithm
- EA set-up per deme

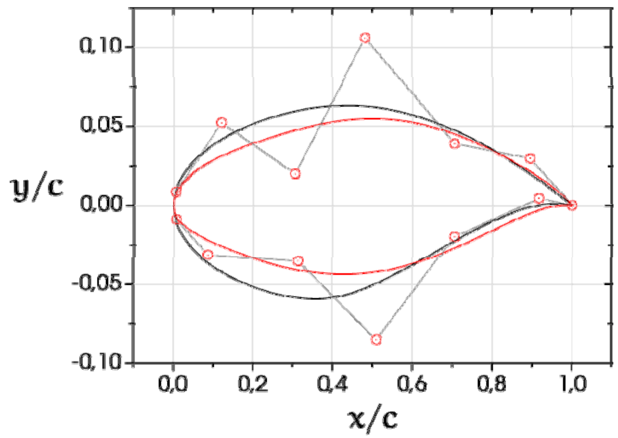
M.K. KARAKASIS, A.P. GIOTIS and K.C. GIANNAKOGLOU: 'Inexact Information Aided, Low-cost, Distributed Genetic Algorithms for Aerodynamic Shape Optimization', Int. J. for Numerical Methods in Fluids, Vol. 43, pp. 1149-1166, 2003.

DMAEA vs. EA or DEA or MAEA– An Example

Airfoil Shape Optimization (min. C_D, fixed C_L)

M_∞=0.75, α_∞ =2.734°, Re=6.2 10^6 **β=2**





Similar behavior can be found in many other cases! A well-tuned DMAEA constantly outperforms other variants, such as EAs, DEAs or MAEAs. This gain doesn't depend on the use of a multi-processor system.

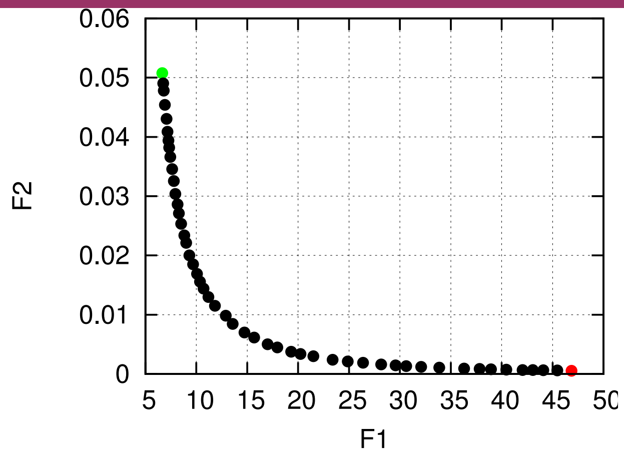
EAs/MAEAs: Curse of Dimensionality

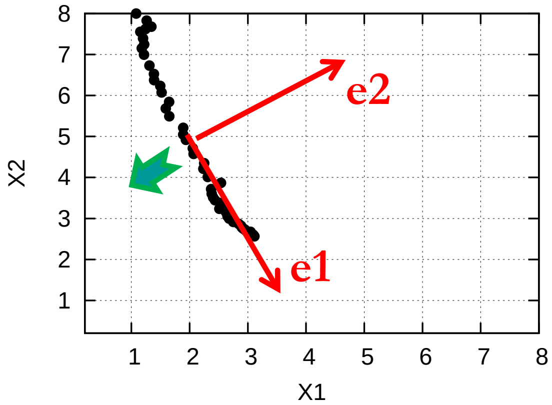
- Industrial optimization problems usually involve a great number of design variables. This causes a serious performance degradation due to:
  1. The (otherwise well performing) evolution operators become inefficient and noticeably increase the number of evaluations required to capture the optimal solution(s).
  2. Metamodels (surrogate evaluation models; irrespective of their type) become less dependable and lead to less accurate predictions.

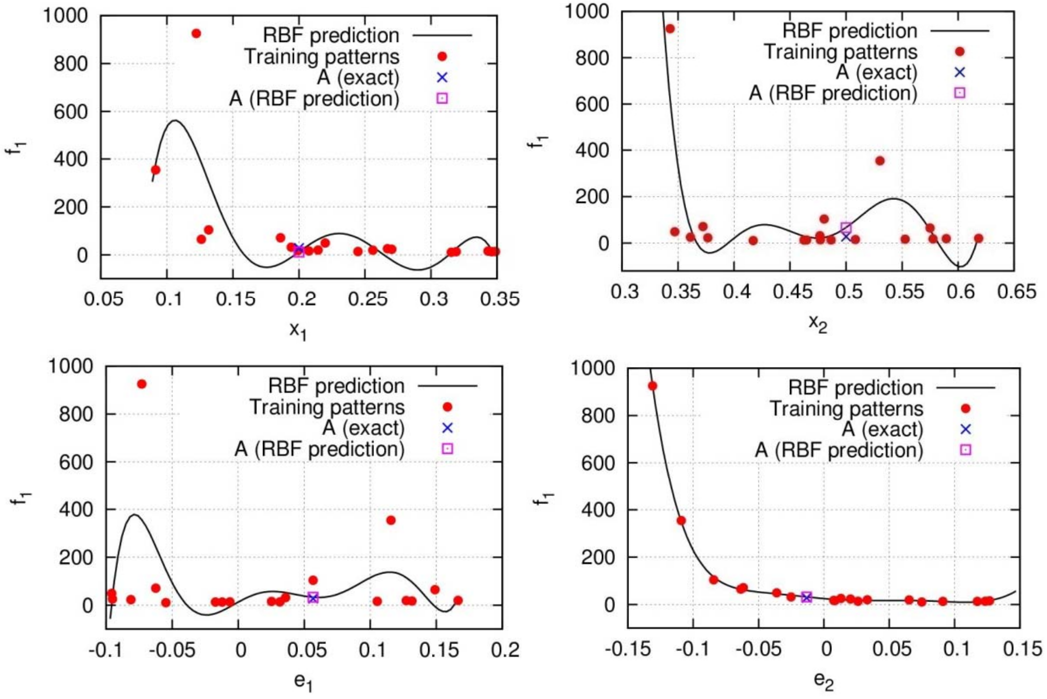
- How to overcome this problem: filter the less important design variables out and apply the evolution operators and/or train metamodels on candidate solutions of lower dimension. A sort of sensitivity analysis is required. In EASY, this is based on the Principal Component Analysis (PCA) of the current best individuals (dynamically updated during the evolution).

S.A. KYRIACOU, V.A. ASOUTI, K.C. GIANNAKOGLOU: 'Efficient PCA-driven EAs and Metamodel-Assisted EAs, with Applications in Turbomachinery', Engineering Optimization, 46(7), 895-911, 2014.

EAs/MAEAs: Curse of Dimensionality







A two-objective (min F_1, min F_2) and two opt. variables (x_1, x_2).

(a) During the application of the evolution operators: Don't care about e_1!!

(b) During the metamodel training: It is much better to train the metamodel only on e_2, namely the "new" design variable with the smallest variance, than on x_1 and x_2.

From EA(PCA) and M(PCA)AEA to M(PCA)AEA(PCA).

Design-Optimization of a Francis Runner at 3 OPs

<u>Objectives (at 3 operating points):</u>

f_1 : exit velocity profiles' quality

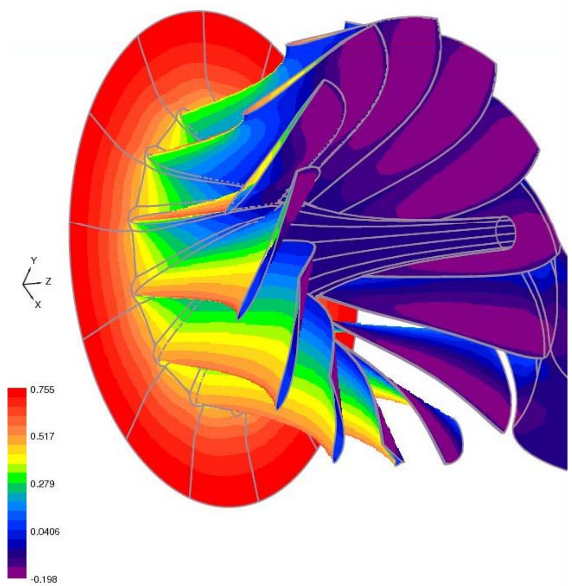
f_2 : uniformity of the blade loading

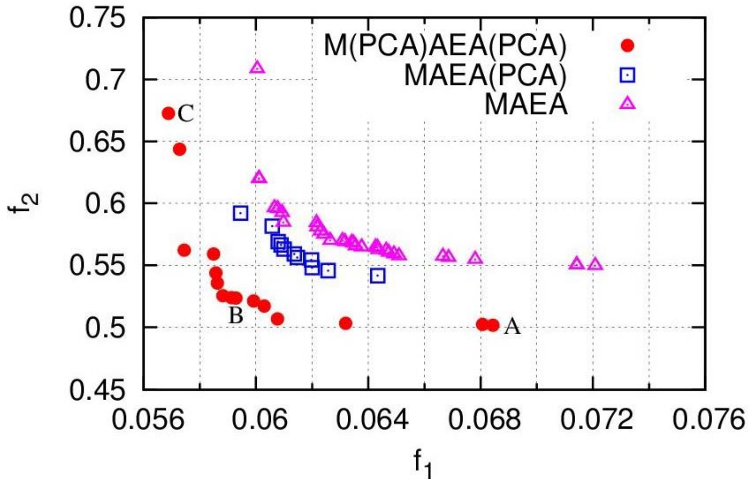
<u>Constraints:</u>

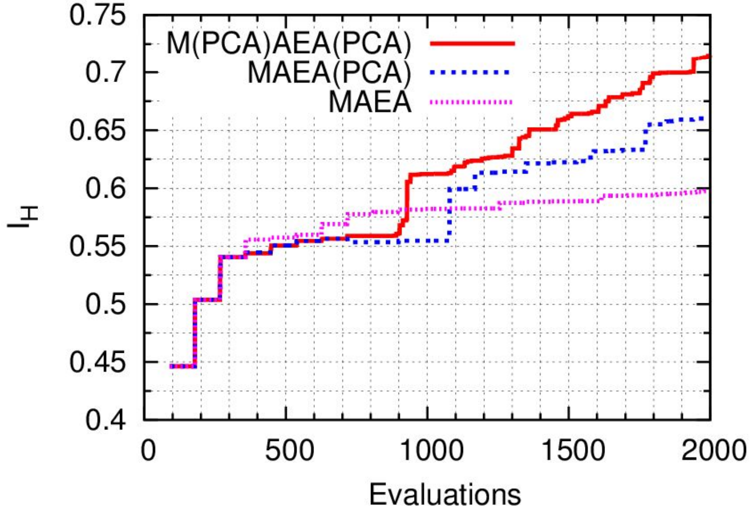
Maintain a pre-defined head

Avoid cavitation

372 design variables







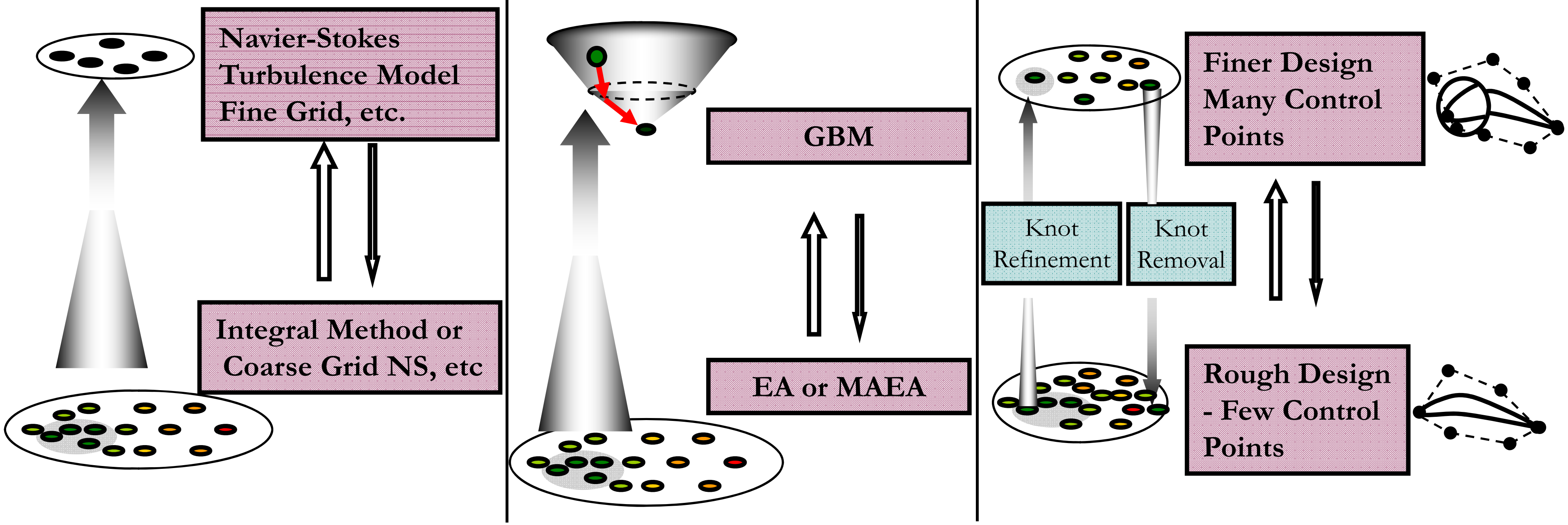
Comparison of fronts of non-dominated solutions obtained at the same number of exact evaluations (same CPU cost).

Hierarchical / Multilevel Optimization Schemes

Splitting the optimization at different (usually two!) levels, allows the combined use of evaluation models of different accuracy and computational cost, different search methods (EAs, MAEAs, DMAEAs, etc. or gradient-based ones) and/or different sets/subsets of design variables.

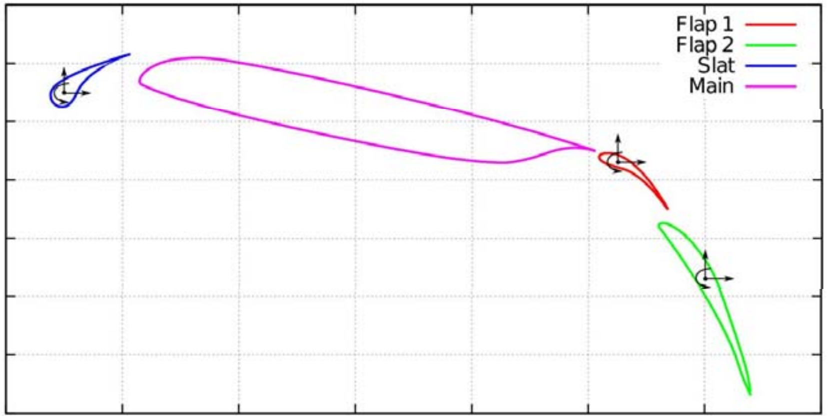
Exploitation vs. Exploration-based levels.

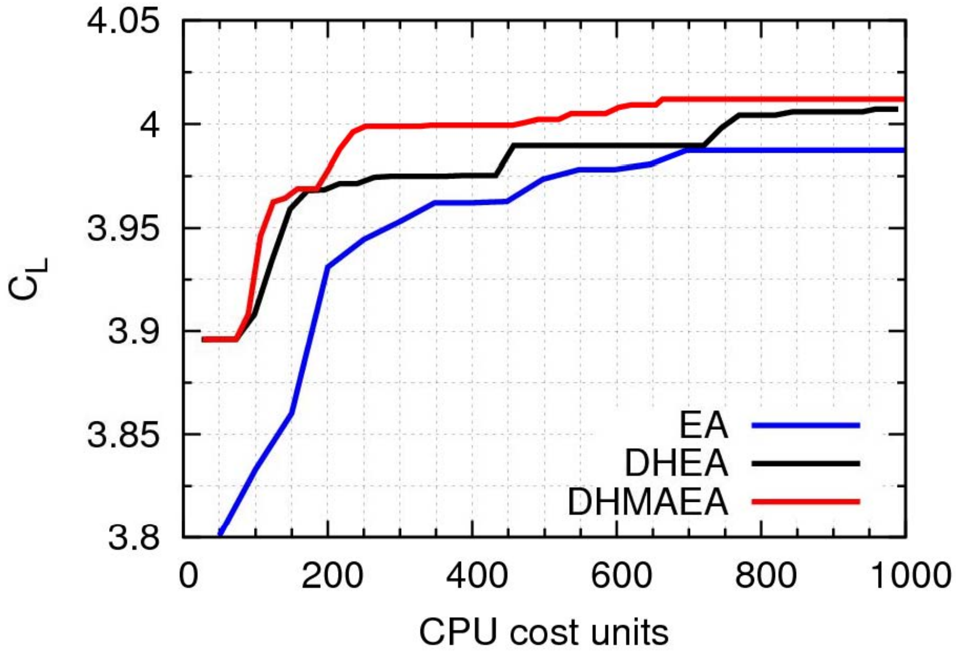
Hierarchical Evaluation	Hierarchical Search	Hierarchical Parameterization



Hierarchical Optimization

Optimal Deployment of a 4-Element Airfoil.
Target: max Lift. - 9 design variables.



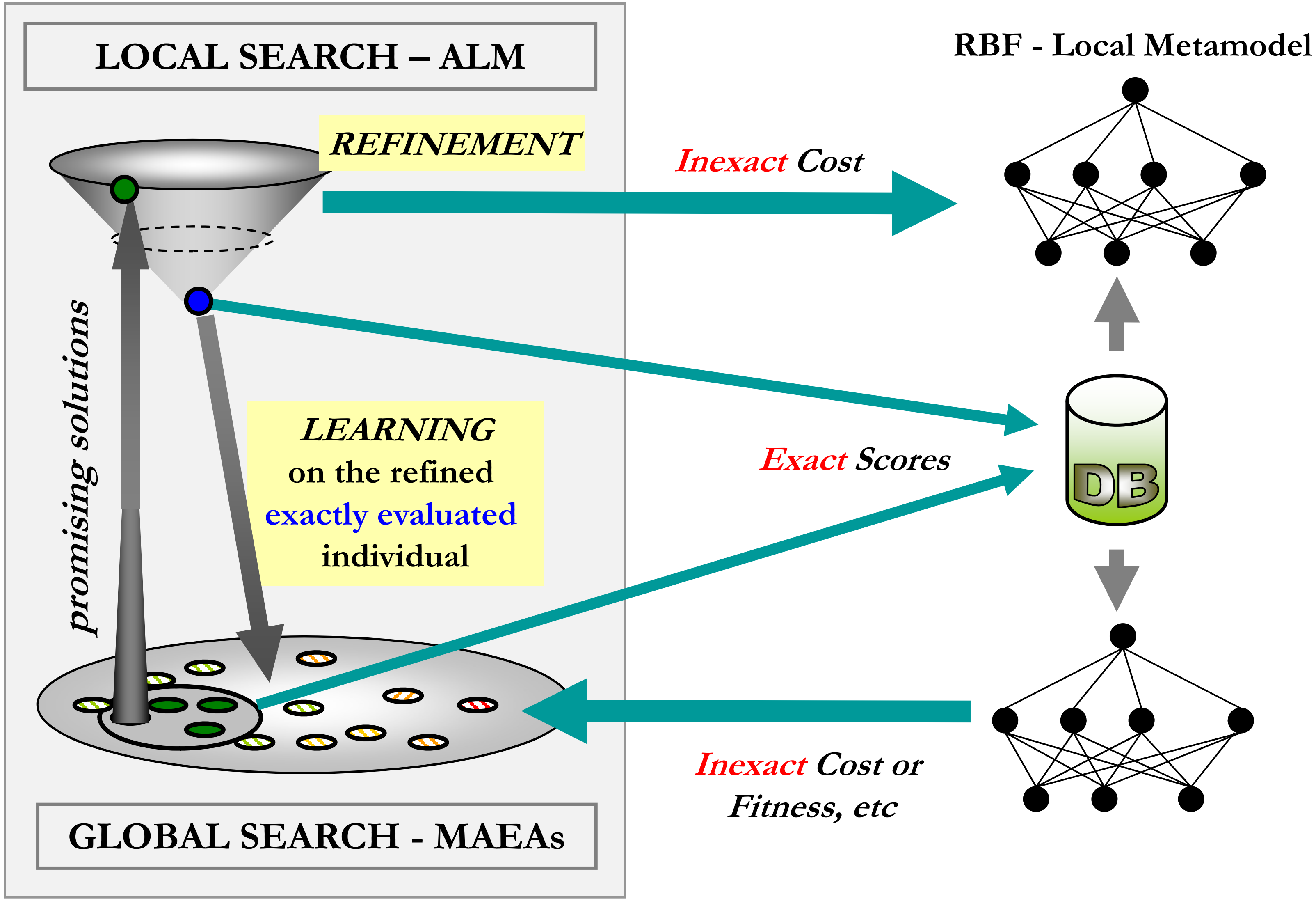


An hierarchical EA was devised by combining single- & double-precision GPU-enabled Navier-Stokes solver (SPA & DPA, respectively).

Comparison of:

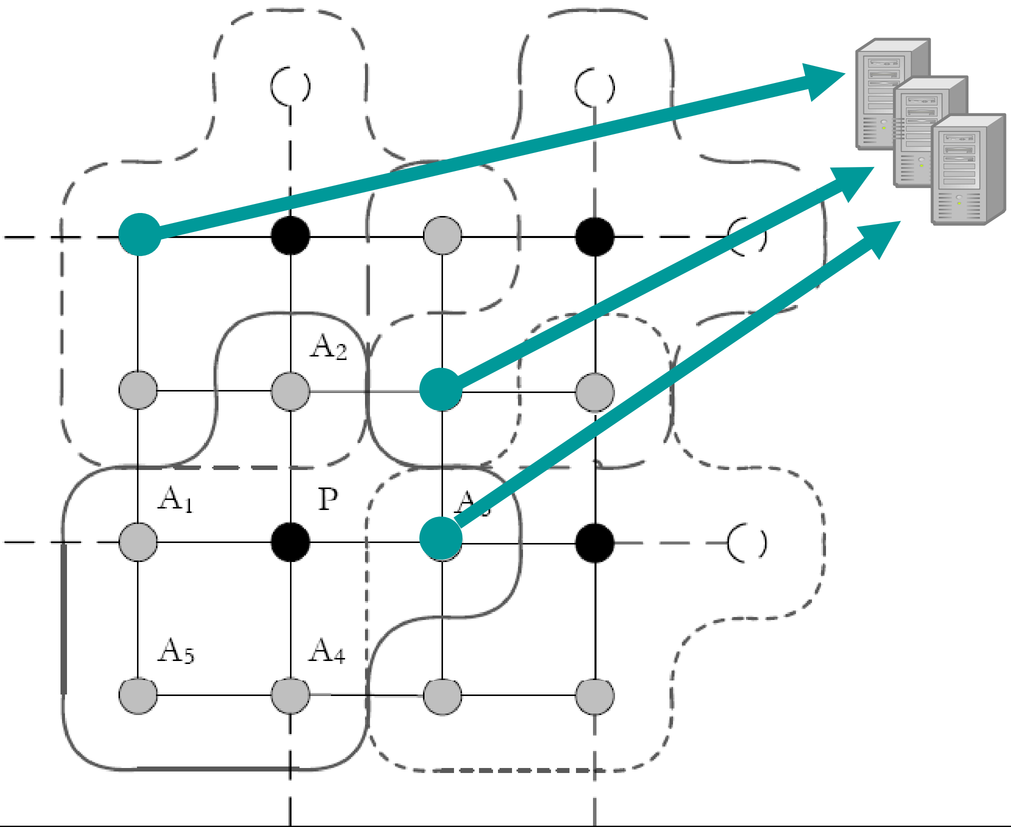
(EA)= (20,50)EA using DPA
(DHEA)= 3 demes x(5,15)EA using SPA for the λ=15 offspring & DPA for the best 2 generation members
(DHMAEA)= 3 demes x(5,15)EA using RBF-metamodel for the λ=15 offspring (IPE), 9 best re-evaluated on SPA & 1 best re-re-evaluated on DPA

Metamodel-Assisted Memetic Algorithms (MAMAs)



C.A. GEORGOPOULOU, K.C. GIANNAKOGLOU: 'A multi-objective metamodel-assisted memetic algorithm with strength-based local refinement', Engineering Optimization, Vol. 41, pp.909-923, 2009.

Asynchronous EAs (AEAs)



Start Algorithm

Receive

- Pole Displacement
- Update Ages
- Update Priorities

$$Pr_p = Pr_p^{age} Pr_p^{cost}$$

- Select new deme
- Select new agent
- Recombine & Mutate

$$x_a = x_p + \omega_r (x_{k_1} - x_{k_2})$$

- Assign new evaluation

The AEA, controlled by strongly interacting demes ensures max. exploitation of all available processors. At the end of an evaluation, the next agent to undergo evaluation on the idle processor is determined by of intra- and inter-deme processes.

V.G. ASOUTI, K.C. GIANNAKOGLOU: 'Aerodynamic optimization using a parallel asynchronous evolutionary algorithm controlled by strongly interacting demes', Engineering Optimization, Vol. 41, pp. 241-257, 2009.

Asynchronous MAEAs (AMAEAs)

Algorithm 1 : *IPE* within the *AMAEA*.

if *IPE* is Activated **then**
  generate N_{IPE} trial members and inexact pre–evaluate them
  for all $t \in \{1, \ldots, N_{IPE}\}$ **do**
    $\mathbf{x}^t_{agent} \leftarrow Recombine\&Mutate()$ /* $\mathbf{x}^t_{agent}$: *trial member* */
    $RBF\ network \leftarrow Train()$ /* *train a local* RBF *network* */
    $\phi^{IPE}_t(\mathbf{x}) \leftarrow UseRBF\ network()$ /* $\phi^{\mathrm{IPE}}_t(\mathbf{x})$: *approximate cost of* $\mathbf{x}^t_{agent}$ */
  end for
  $\mathbf{x}^{new}_{agent} \leftarrow Select(\phi^{IPE}_t(\mathbf{x}))$ /* *select the best according to the* RBF *network* */
end if

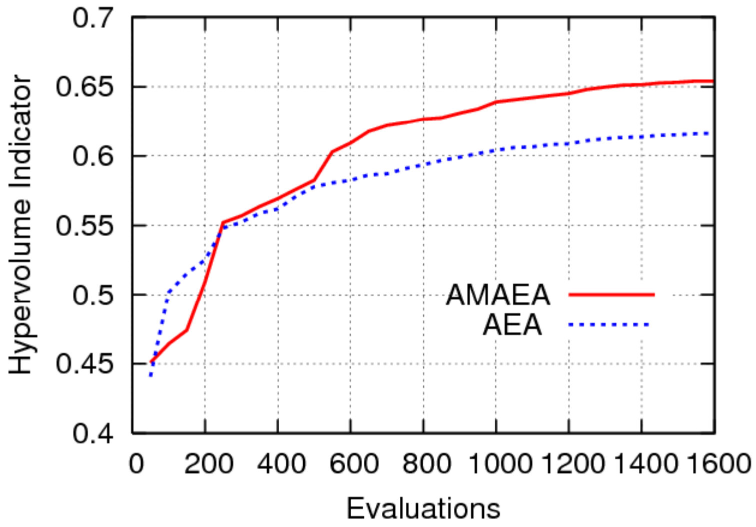
After the decision on the next agent to undergo evaluation is made, instead of generating a single new member to be evaluated, N_{IPE} trial members are generated. For each trial member, a local metamodel is trained. The most promising new individual among the N_{IPE} ones, according to the metamodel (used separately for all of them), is re-evaluated on the CFD code.

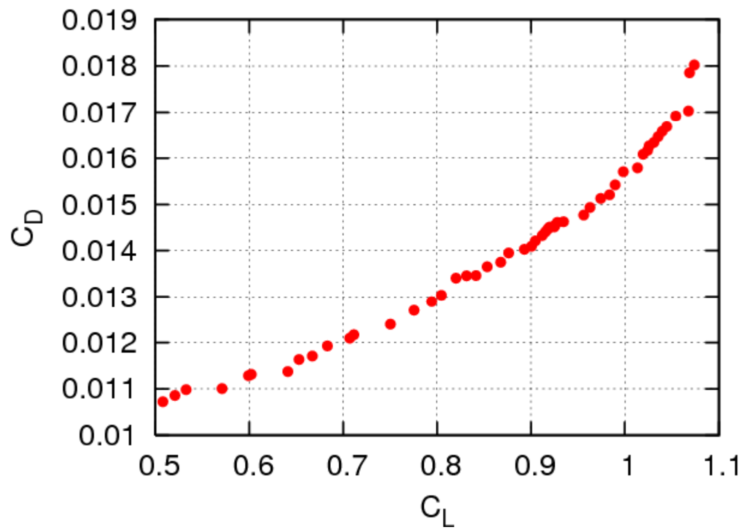
V.G. ASOUTI, I.C. KAMPOLIS, K.C. GIANNAKOGLOU: 'A Grid-Enabled Asynchronous Metamodel-Assisted Evolutionary Algorithm for Aerodynamic Optimization', Genetic Programming and Evolvable Machines, Vol. 10, No. 4, pp. 373-389, 2009.

Asynchronous EAs/MAEAs

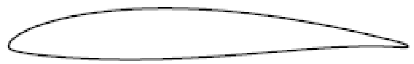
Design of an isolated airfoil for min CD/max CL (MOO) :

Flow Conditions M_∞=0.3, α_∞ =4^o, Re=5 10^6





$C_L = 0.508, C_D = 0.0107$



$C_L = 0.9047, C_D = 0.0142$

$C_L = 1.0739, C_D = 0.018$

Summary/Conclusions

- Evolutionary optimization may profit a lot from the use of approximate, computationally cheap metamodels which can replace the exact and costly problem–specific evaluation code, for a great number of non–promising candidate solutions.
- Metamodels can also be used in hierarchical or mulitlevel optimization schemes, in which at least one of the levels relies on EAs. The gain offered by using metamodels is superimposed to that of using more than one optimization levels.
- In EAs or MAEAs, the use of PCA-driven "filters" is a remedy to the increased computational cost caused by the curse of dimensionality.
- Any implementation of local search within an EA, i.e. according to the so–called memetic algorithms, may also profit of the use of metamodels (not shown in detail here).
- Asynchronous search methods, which are appropriateCluster and Grid Computing, may also use metamodels, though in a different way than before.

Όλοι οι υπολογισμοί έγιναν με το λογισμικό EASY που έχει αναπτυχθεί στο ΕΜΠ.

The Evolutionary Algorithm SYstem

http://velos0.ltt.mech.ntua.gr/EASY

http://147.102.55.162/EASY