



National Technical University of Athens
School of Mechanical Engineering
Department of Fluids
Lab of Thermal Turbomachines
Parallel CFD & Optimization Unit

Evolutionary Algorithm-based Design-Optimization Methods in Turbomachinery

A Thesis Submitted to the National Technical University of Athens for the
degree of Doctor of Philosophy in the Faculty of Mechanical Engineering

Stylianos A. KYRIACOU

Supervisor : KYRIAKOS C. GIANNAKOGLOU
Professor NTUA

Athens, 2013

Abstract

The scope of this PhD thesis is to propose a set of improvements to existing shape design-optimization methods in fluid dynamics based on Evolutionary Algorithms (EAs) and demonstrate their efficiency in real-world applications. Though the proposed method and the developed EA-based software are both generic, this thesis focuses on applications in the fields of hydraulic and thermal turbomachines. With the proposed algorithmic variants, the optimization turn-around time is noticeably reduced with respect to that of conventional (reference, background) methods. Though the latter are computationally expensive, with the proposed add-ons, they become affordable even for large-scale industrial applications. The background design-optimization methods are based on EAs enhanced by the use of artificial neural networks, acting as surrogate evaluation models or metamodels. The design process is coupled with the necessary Computational Fluid Dynamic (CFD) software. Parallelization, in the form of concurrent evaluations of candidate solutions within each generation of the EA, is absolutely necessary, given the high CPU cost per CFD-based evaluation. All computations were performed on the multi-processor platform of the Parallel CFD & Optimization Unit (PCOpt) of the National Technical University of Athens (NTUA). Using the proposed optimization methods several turbomachinery design optimization problems are solved; these include the design-optimization of traditional hydraulic turbines (such as Francis turbine) and a new/innovative variation of bulb turbines, the so-called “Hydromatrix®”, suitable for low head hydropower sites. These computations were performed in close collaboration with a major hydraulic turbine manufacturer (Andritz Hydro). Furthermore, the compressor cascade installed at the Lab of Thermal Turbomachines of NTUA (LTT/NTUA) is optimized. This thesis presents also the application of the proposed methods and tools on a number of mathematical optimization problems, since these allow a great number of test runs to be performed at negligible CPU cost.

The most important contributions of this thesis are listed below:

- a) A new design method, which fully exploits archived designs with good performance in “similar” conditions, is proposed. In this method, EAs (either in their conventional form or in enhanced variants assisted by metamodels) solve a reformulated optimization problem, instead of the high-dimensional real one. The new unknowns are used to non-linearly weight the archived designs so as to create candidate solutions to the problem in hand; it is a great advantage of the proposed method that, by doing so, the EA avoids handling the, otherwise, great number of variables parameterizing the complex shape to be designed. Gains from the reduction of the number of design variables are evident. An extra gain in CPU cost arises from the statistical analysis of the archived designs that helps identifying the most important design space regions, where greater probabilities of accommodating new offspring are given. The proposed method has the additional advantage of allowing the automatic definition of the new variables’ upper and lower bounds.
- b) The use of principal component analysis (PCA) as a means to reveal the topological characteristics of the current generation elite set is proposed. The principal directions on the design space, as computed via PCA in each generation, are used for the rotation of the design variable coordinate system before the application of the evolution operators. This rotation suffices to transform an ill-posed optimization problem into a better-posed one. Through the proposed PCA-driven evolution operators, offspring generations of higher quality are created and, this certainly reduces the number of generations required to get the optimal solution(s). Note that this is done without truncating the design variables, as a few other method are doing.
- c) The same PCA technique, along with the information about the design variable importance is used to enhance the inexact pre-evaluation (IPE) phase in metamodel-assisted EAs (MAEAs). This is, herein, applied in MAEAs incorporating radial basis function (RBF) networks as metamodels, in either single- or multi-objective optimization problems. The proposed method aims at obtaining more relevant predictions of the objective function values by the online trained metamodels. To this end, the PCA-driven rotation of the design variables followed by an appropriate truncation of the less-significant among them allow the use of metamodels trained on patterns of smaller dimension. Consequently, the training can rely on a smaller number of patterns and evaluations on the RBF network are much better. The EA based search is, thus, better driven and this leads to lower CPU

costs.

The gain from the use of the proposed methods, either separately or in combination, is quantified in a few mathematical test cases and, then, some real-world applications. These were solved twice, using both the proposed and the background optimization methods. In more detail, the gain from the use of KBD is demonstrated by solving the design problem of a 3D Francis hydraulic-turbine and that of a 2D compressor cascade. The use of an EA with PCA-driven evolution operators is shown in the design-optimization of a “Hydromatrix®”. The use of PCA to enhance the performance of metamodels in MAEAs is demonstrated in the design of a 2D compressor cascade and the optimization of the blade shape of a peripheral compressor cascade installed and measured at LTT/NTUA.



National Technical University of Athens
School of Mechanical Engineering
Department of Fluids
Lab of Thermal Turbomachines
Parallel CFD & Optimization Unit

Evolutionary Algorithm-based Design-Optimization Methods in Turbomachinery

Doctoral Dissertation
STYLIANOS A. KYRIACOU

Examination Committee:

1. K. Giannakoglou (Supervisor)*
Professor, NTUA, School of Mechanical Engineering.
2. K. Mathioudakis*
Professor, NTUA, School of Mechanical Engineering.
3. J. Anagnostopoulos*
Associate Professor, NTUA, School of Mechanical Engineering.
4. D. Papantonis
Professor, NTUA, School of Mechanical Engineering.
5. G. Grigoropoulos
Professor, NTUA, School of Naval Architecture and Marine Engineering.
6. L. Kaiktsis
Associate Professor, NTUA, School of Naval Architecture and Marine Engineering.
7. N. Aretakis
Lecturer, NTUA, School of Mechanical Engineering.

* Member of the Advisory Committee.

Athens, 2013

to thirteen

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Professor Kyriakos Giannakoglou, Lab of Thermal Turbomachines and Head of the Parallel CFD & Optimization Unit, School of Mechanical Engineering, National Technical University of Athens. His wide knowledge and structured way of thinking have been of great value for me. His understanding, encouraging and personal guidance have provided a good basis for the present thesis.

I also would like to thank Professor K. Mathioudakis and Associate Professor J. Anagnostopoulos for their valuable comments and participation in the advisory committee of my PhD thesis.

I am deeply grateful to Erwin Oberbichler, Simon Weissenberger and Peter Grafenberger (Andritz Hydro Linz) for guiding me into the inner workings of hydraulic turbines and, above all, sharing their experience and knowledge on what the optimal turbine should really look like. I also wish to thank all of my colleagues in Andritz Hydro Linz for their detailed and constructive comments, and for their important support throughout this work.

I wish to express my warm and sincere thanks to Arno Gehrer (Andritz Hydro Graz) for sharing his ideas and concepts on optimization regarding hydraulic pumps and pump-turbines and Irida Skouteropoulou (Andritz Hydro Graz) and Kostas Tsakas (ex Andritz Hydro Graz and, currently PhD student at NTUA) for implementing and testing them in collaboration with the optimization methods developed in this PhD thesis. Furthermore I would like to thank Etienne Parkinson and Florent Champet (Andritz Hydro Vevey) for their ideas concerning optimization of pelton distributors.

My warm thanks are due to Paul Pieringer and Livia Koch for providing the parameterization, grid generation and flow solver tools used for the industrial-scale hydraulic turbine optimization problems solved in this thesis.

I warmly thank Dr. Ioannis Kampolis for laying the stable foundations on which this thesis was built upon and guiding me during my early

research years. Also, I would like to thank Dr. Varvara Asouti for her valuable help in the field of Evolutionary Algorithms.

During this PhD thesis, I have collaborated with many colleagues, from the PCOpt/NTUA, for whom I have great regard. Without exception, all new ideas/methods presented in this thesis have their roots in constructive conversations with them and for that I warmly thank them.

Also, I would like to thank Peter Kunz, Christian Fuereder, Gabriele Humer and Markus Oswald for the construction and experimental validation of the Hydromatrix® turbine presented in Appendix 2.

Last but not least, I owe my loving thanks to my family, Antonis, Anna and Nektaria Kyriacou. Without their encouragement and understanding, it would have been impossible for me to finish this PhD thesis.

The financial support of Andritz Hydro and the EU project named “HYDROACTION-Development and laboratory testing of improved action and Matrix hydro turbines designed by advanced analysis and optimization tools” (PROJECT NUMBER : 211983) is gratefully acknowledged.

Nomeclature

CFD	Computational Fluid Dynamics
EA	Evolutionary Algorithm
HEA	Hierarchical Evolutionary Algorithm
IPE	Inexact Pre-Evaluation
KBD	Knowledge Based Design
PCA	Principal Component Analysis
MAEA	Metamodel-Assisted Evolutionary Algorithm
MAEA(PCA)	MAEA with PCA-Driven Evolution Operators
M(PCA)AEA(PCA)	MAEA(PCA) with PCA-Driven Metamodels
TE	Trailing Edge
LE	Leading Edge
PL	Part Load
FL	Full Load
BE	Best Efficiency

Contents

1	Introduction	1
1.1	CFD-Based Optimization	3
1.2	EAs: Previous work at PCOpt/NTUA	9
1.3	Thesis Outline	10
2	Evolutionary Algorithms (EAs)	13
2.1	Introduction - Overview of EAs	13
2.2	Optimization Problems - Definitions	18
2.2.1	Multi-Objective Optimization and EAs	18
2.2.2	Constrained Optimization and EAs	19
2.3	The Evolutionary Algorithm SYstem (EASY)	20
2.3.1	The (μ, λ) EA	21
2.3.1.1	Evolution Operators	22
2.3.2	Solving MOO Problems with EASY	29
2.3.3	Constraints' Handling in EASY	31
2.3.4	Metamodel-Assisted Evolutionary Algorithm (MAEA)	32
2.3.4.1	Inexact Pre-Evaluation Algorithm	35
2.3.5	Hierarchical Evolutionary Algorithms	36
3	Knowledge-Based Design	39
3.1	Knowledge-Based Systems (KBS)	40
3.1.1	Case-Based Reasoning Method - Principles	41
3.2	The KBD Method	42
3.2.1	Optimization Variables's Definition	43
3.3	Design of a Compressor Cascade using KBD	46
4	EAs and MAEAs Driven by Principal Component Analysis	53
4.1	Ill-Posed Optimization Problems	54
4.1.1	Anisotropic Objective Function	55
4.1.2	Non-Separable Objective Function	56

CONTENTS

4.1.3	Investigation of EAs Efficiency Degradation in Ill-Posed Problems	57
4.2	Ill-Posed MOO Problems	61
4.3	Principal Component Analysis	61
4.4	The Welded Beam Case	62
4.5	PCA-Driven Evolution Operators	64
4.5.1	PCA-Driven Recombination	66
4.5.2	PCA-Driven Mutation	67
4.5.3	Gain from the Use of EA(PCA)	68
4.6	EAs Assisted by PCA-Driven Metamodels	69
4.6.1	Gain from the Use of M(PCA)AEA(PCA)	74
4.7	Design of a Compressor Cascade	76
5	Optimization of Hydraulic Turbomachines	79
5.1	Evaluation Procedure	81
5.1.1	Parameterization for Turbine Blades	81
5.1.2	Space Discretization - Grid Generation	84
5.1.3	Flow Solver	87
5.2	Hydraulic Turbine-Runner Quality Metrics	88
5.2.1	Cavitation Metric	88
5.2.2	Outlet Velocity Profiles	90
5.2.3	Blade Loading Quality Metric	91
5.2.4	Pumping Surface Metric	93
5.3	Optimization of a Francis Turbine Runner	94
5.3.1	Francis Turbine	95
5.3.2	Case Presentation	96
5.3.2.1	Design Parameterization	97
5.3.2.2	Objectives and Constraints	98
5.3.3	Results	100
5.3.3.1	Archived Designs	100
5.4	Optimization of a Hydromatrix® Turbine	119
5.4.1	The Hydromatrix® Turbine	119
5.4.2	Case Presentation	121
5.4.2.1	Case Formulation	121
5.4.2.2	Design Parameterization	121
5.4.2.3	Objectives and Constraints	123
5.4.3	Results	124
6	Optimization of a Compressor Annular Cascade	135
6.1	Blade Shape Parameterization	136
6.2	Case Presentation	137

CONTENTS

6.2.1	Objectives and Constraints	137
6.3	Results	139
7	Conclusions - Future Work	143
7.1	Future Work	145
7.2	Publications - Conference Presentations	145
A	Regulated Hydraulic Turbines	147
A.1	Single-Regulated Turbines	147
A.2	Double-Regulated Turbines	149
B	Design of a Hydromatrix®: Experimental Validation	151
B.1	Optimization Procedure	151
B.2	Experimental Validation	155
	Bibliography	159

CONTENTS

1

Introduction

In this PhD thesis, new Evolutionary Algorithm (EA) variants for solving large-scale design-optimization problems at reasonable computing cost are proposed, programmed and validated. This is extremely useful when handling large-scale industrial optimization problems, such as those encountered in the fields of thermal and hydraulic turbomachines. The developed EA variants, enhanced with the proposed add-on features, must be able to produce high quality designs, at acceptable (according to the industrial standards) turn-around time. The latter is, in fact, necessary for EAs to become routinely used design-optimization tools in industrial environment, where the search method has to cope with a usually great number of design variables and computationally expensive evaluation codes.

A clear understanding of the reasons causing the drop in EA's efficiency when used to solve industrial problems, particularly large-scale ones, is necessary before proceeding to the presentation of the proposed remedies. In this thesis, the problems under consideration are usually referred to as "ill-posed". They are dealing with anisotropic (section 4.1.1), non-separable (section 4.1.2) objective functions and have a great number N of design or optimization variables. The solution to an optimization problem with a non-separable objective function, as opposed to a separable one, cannot be obtained by solving N distinct optimization problems, one for each design variable, since the optimal value of each one of them depends on the values of the rest. In terms of N , this leads to superlinear increase in computational cost, in contrast to the linear increase expected in separable problems. It is evident that, for this type of problems, particularly if N is quite high, strong retardation in EAs convergence is expected. Note that the majority of industrial-scale problems are non-separable with a great number of unknowns.

The first way proposed in this thesis for reducing the optimization turn-around time relies on the replacement (during the EA-based search) of the standard shape parameterization techniques, which introduces a great number of design variables, by resorting to a number of available satisfactory designs. In fact,

1. Introduction

this thesis proposes and evaluates a way to use the information residing into a small number of archived designs, made available from similar successful projects worked out in the past. This leads to optimization problems with much less unknowns which can be solved using EAs at noticeably lower computational cost. The new optimization variables are the coefficients introduced for merging the existing designs, in order to create new ones. The proposed method combines “ideas” from the theory of Knowledge-Based Systems (KBS) with EAs, giving rise to a fast design method, which will be referred to as Knowledge-Based Design (KBD). Though the proposed KBD method does not explicitly take care of the non-separability of the objective function with respect to the design variables, it is important that the number of optimization variables is greatly reduced. So, even without explicitly coping with the non-separability property, the gain in CPU cost is noticeable.

The second method proposed herein makes use of the Principal Component Analysis (PCA) of promising/top individuals, in each generation of the EA, in order to identify directions in the design space which, if used to redefine the optimization variables, would result in “better-posed” optimization problems. The redefinition of the optimization variables comes out from the alignment of each design vector with the so-called principal directions, computed by the PCA. The evolution operators are applied to the transformed optimization variables and, for reasons to be explained as the text develops, an efficient search mechanism results, even without dimensionality reduction.

Both the KBD method and the new way to apply the evolution operators driven by the PCA analysis of the current most promising solutions (such as the current front of non-dominated solutions in multi-objective optimization problems) can be used within either an EA or a Metamodel-Assisted EA (MAEA). In MAEAs, the costly problem-specific evaluation tool (herein, a Computational Fluid Dynamics or CFD code) is replaced by low-cost surrogate evaluation models or metamodels (trained artificial neural networks, ANNs, polynomial regression methods, etc.). In each generation, the CFD code is used to “exactly” evaluate a few of the most promising population members, as pointed by the metamodel(s).

Regarding MAEAs, another way to further enhance their efficiency is by using the same PCA of the current promising solutions in order to associate each design variable (or design space coordinate) with a degree of importance. It will be shown that MAEAs may benefit a lot from the importance-based ranking of the design space coordinates, in order to overcome a well-known problem caused by the so-called curse of dimensionality. This problem is related to the fact that, for the ANNs used to approximate the fitness or cost of a new individual, any increase in the number of their sensory units calls for more training patterns and increased training cost. Since the training patterns are selected among the previously evaluated individuals during the EA, the need for more training patterns

means that the use of metamodels must be delayed. If, on top of this, each training is more costly, the gain from employing metamodels in lieu of the CFD code is expected to be lower or may even vanish. The proposed method is based on the selective truncation of the ANN entries, by maintaining only the most important design variables according to the results of PCA. It will be shown that, in real-world applications, the gain is important.

The aforementioned methods are, firstly, validated in cases ranging from some low-cost mathematical optimization benchmarks to 2D compressor cascade designs. Then, real-world problems such as the design of industrial 3D hydraulic turbines and the redesign of the 3D annular compressor cascade with tip clearance installed at the Lab of Thermal Turbomachines of NTUA (LT/NTUA) are examined. The mathematical benchmarks have low computational cost and allow the exhaustive investigation of methods by repeating the runs on the same case, using different seeds in the random number generator. By doing so, safe conclusions are reached. The presented 2D cases allow the demonstration of the proposed methods in simple aerodynamic cases. In the field of hydraulic turbomachines, two types of hydraulic turbines, a Francis and a new type named Hydromatrix®, are used to validate the performance of the aforementioned methods in large-scale industrial applications. A number of performance metrics are introduced and, depending on the case, combined to form the necessary objectives and constraints. In the hydraulic turbomachinery design problems, more than one operating points are considered, leading to increased cost per evaluation. This makes the reduction in the number of evaluations required to reach the optimal solution(s) absolutely necessary. In the field of thermal turbomachines, the blade airfoil of the 3D compressor cascade installed at LTT/NTUA is optimized for minimum total pressure losses.

This PhD thesis is based on EAs developed during a number of previous PhDs (15, 74, 92, 98) performed at the Parallel CFD & Optimization Unit of LTT/NTUA (PCOpt/NTUA). These PhD theses created the algorithmic basis or software platform, which the newly proposed methods are built upon. This EA-based optimization platform, known as EASY (Evolutionary Algorithm SYstem (70)) includes the basic EA, optionally enhanced by parallel search (Parallel EAs or PEAs), the use of ANNs as metamodels (MAEAs) and hierarchical optimization schemes (Hierarchical EAs or HEAs). EASY provided fertile ground for the development of all methods proposed in this PhD thesis.

1.1 CFD-Based Optimization

In this section, an overview of CFD-based optimization methods is presented. CFD-based optimization methods enjoy great interest from both academia and

1. Introduction

industry since the performance and, therefore, the price of products (ranging from cars and aircrafts to thermal and hydraulic turbomachines etc.) depends heavily upon their aero/hydrodynamic performance.

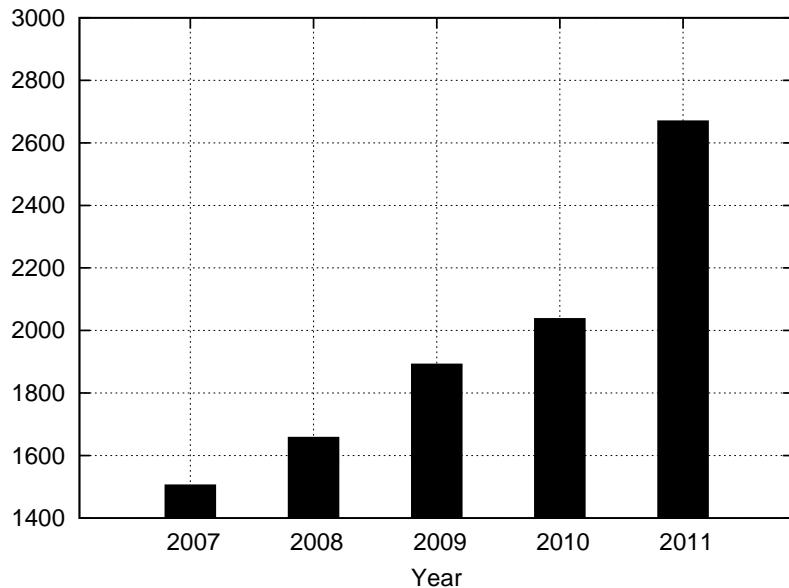


Figure 1.1: Yearly number of publications on CFD-based optimization appearing in ScienceDirect.

In order to demonstrate the increasing interest of the scientific community in CFD-based optimization, an internet-based literature survey was carried out using the search tool “ScienceDirect” (<http://www.sciencedirect.com/>). The search lemma was (Optimization *AND* CFD) *OR* (Optimization *AND* Computational Fluid Dynamics). This is obviously a non-exhaustive search; it is though good enough to show trends. The steadily growing number of publications in the last five years, shown in fig. 1.1, reveals the growing interest in performing CFD-based optimization. Therefore, it is expected that the upgrades proposed in this thesis must be of interest to a continuously growing in size community of scientists, not necessarily restricted to those working in the field of turbomachines.

A prerequisite for any shape optimization problem in fluid dynamics is the availability of a fast and reliable way to evaluate candidate solutions. This is achieved through the use of CFD (hence the term CFD-based optimization) methods that numerically solve the differential equations governing the fluid motion. Nowadays, thanks to the relatively cheap access to powerful computational means, the CFD tools have affordable CPU cost (depending on the case complexity) and, therefore, optimization methods based on them are flourishing. The strong dependence of the CFD-based optimizations on the computational resources be-

1.1 CFD-Based Optimization

comes crucial in cases where several disciplines (ranging from structural analysis to manufacturing and economics, etc.) are used to evaluate the candidate solutions (multi-disciplinary optimization, MDO). The fact that the fluid dynamic, structural and economical objectives may be contradictory adds extra difficulty to the optimization procedure.

An important part of any optimization project is the definition of the design variables. An optimization algorithm seeks the value set of the design variables which minimizes the user-defined cost function(s). In CFD-based optimization, typical objectives are: maximization of lift, minimization of drag, maximization of efficiency, minimization of the deviation from given target (pressure, velocity, etc.) distributions, minimization of viscous or shock-induced losses or the cavitation index (in hydraulic turbomachines), etc. A single objective (single-objective optimization, SOO) or combinations of them can be used. Design-optimization problems with more than one objectives (multi-objective optimization, MOO) can be solved by concatenating the objectives into a single one, after multiplying them with appropriate weights. The difficulty of choosing the most appropriate weight values and the failure of this approach in non-convex optimization problems are the main weaknesses of this approach. An improved way to solve MOO problems is the use of dominance-based methods that rank the candidate solutions according to Pareto dominance¹. These methods compute a set of optimal solutions forming the so-called front of non-dominated solutions or Pareto front, where none of them is dominated by any other front member with respect to all objectives. Therefore, the choice of the optimal solution to be adopted relies upon additional criteria set by the decision-maker. The design-optimization problem can, also, be subject to a number of constraints. In shape optimization related to fluid mechanics, constraints are imposed to geometrical quantities (for instance, thickness constraints so as to come up with manufacturable designs), structural quantities such as the maximum stress or deflection and/or flow-related quantities such as the minimum flow turning or the cavitation index, in turbomachines.

In general, CFD-based shape optimization methods can be classified in two categories, namely direct and inverse ones. Direct methods (48, 49, 74, 92, 135) solve the optimization problem through a number of trials that involve geometry generation, CFD-based flow predictions, computation of the objective function and, if necessary, its gradient computation. On the contrary, inverse methods (should not be confused with inverse design problem² that can be solved by direct methods) (25, 37) begin from a set of desirable flow field characteristics (such as those related to the boundary layer) and solve the inverse problem to

¹Solution a dominates solution b if and only if a is equal to b with respect to all objectives but one and better than b at least for one of them.

²Inverse design problems use the deviation of the current from a desirable flow variable distribution as cost function.

1. Introduction

define the geometry. The major advantage of inverse methods is their low cost and the fact that they don't need a starting geometry. On the other hand, their most noticeable drawback is the inability to handle constrained optimization problems. The inverse methods developed by LTT/NTUA are based on the Le Foll method (119) which is an integral method using a two-parametric boundary layer velocity profile. This method was extended to accommodate compressibility (137), surface curvature effects on the turbulent boundary layer development (139) and boundary layer separation (138) and was used for the design of both axial and radial turbomachines. Based on the above classification, this PhD thesis is dealing with direct design-optimization methods.

Based on the way optimal solutions are sought, the optimization methods can be classified as deterministic and stochastic. In CFD-based optimization, deterministic methods were used first, mainly due to their solid mathematical background and fast convergence, to locate optimal solutions with less trials. Deterministic methods require the computation of the gradient of the objective function with respect to the design variables. This limits the type of objective functions such a method might handle to those being differentiable. The cost of the gradient computation might become prohibitive if access to the evaluation software source code is not granted and computationally expensive finite-difference schemes must be employed. On the other hand, in MOO problems, they have difficulties to compute fronts of non-dominated solutions with a single run and they risk to be trapped into local minima.

In the deterministic or gradient-based optimization methods used in fluid dynamics, the most important part is the gradient computation. The use of finite differences, as mentioned above, is not suitable since it requires, at least, as many flow analyses as the design variables. The use of adjoint techniques can reduce this cost to that of a single-equivalent flow solution irrespective of the number of design variables. Due to the need of significant investment in person-months (mathematical development and programming) required for the development of the adjoint methods and software, these were not used in CFD-based optimization since mid-80's (87, 88, 89, 143). The two variants of adjoint techniques, discrete and continuous, differ in the way the discretized adjoint equations are formed. The continuous adjoint method (14, 89, 135) processes the flow PDEs to derive the adjoint ones which are, then, discretized. On the other hand, in the discrete adjoint method, the adjoint equations result directly from the discretized flow equations, (13, 46). A more detailed presentation of adjoint methods used in fluid dynamic optimization can be found in (135).

On the other hand, stochastic optimization methods do not suffer from the risk of getting trapped into local minima due to the probabilistic search they employ. The differentiability of the objective function is not required. An additional advantage of stochastic optimization methods, especially regarding industrial usage,

1.1 CFD-Based Optimization

is their ability to handle any problem, by merely accommodating the available evaluation software, without even having access to its source code. Their main drawback, though, is the need to evaluate a great number of candidate solutions before reaching the optimum. This may significantly increase the overall computational cost of the optimization procedure. This PhD thesis is dealing with stochastic optimization methods and, more precisely, EAs and aims at the reduction of computational cost.

The use of EAs in CFD-based optimization was delayed until mid-90's, mainly due to their high computational cost. The first relevant works (60, 142, 148) presented automated EA-based procedures for aerodynamic design-optimization problems. Soon after, emphasis was laid on the reduction of the cost of the overall optimization procedure, mainly by adapting the available variable representations and evolution operators. This continued with their hybridization with deterministic methods. For instance, in (124) and (56), an EA was used to spot a good starting solution for a deterministic optimization method. In (38), EAs were hybridized with Sequential Quadratic Programming (SQP) for constrained shape optimization problems. Furthermore, the use of distributed EAs (40, 165), surrogate evaluation models (45, 68, 75, 92, 150), hierarchical schemes (39, 44, 92, 108, 166) and concurrent evaluations on a multiprocessor platform (15, 74, 120) have been proposed. Nowadays, the appeal of CFD optimization using EAs for industrial-scale engineering problems is growing due to the availability of powerful computational resources. This PhD thesis is dealing with the use of EAs for solving industrial-scale design-optimization problems in the fields of thermal and hydraulic turbomachines in acceptable (for industry) turn-around time.

Regarding CFD-based optimization in the fields of thermal and hydraulic turbomachines, another internet-based literature survey was carried out, too. Search lemmata were: (a) optimization *AND* thermal *AND* turbomachines and (b) optimization *AND* hydraulic *AND* turbomachines. This survey reveals the steadily growing number of publications, in both turbomachinery areas, over the last five years, as shown in fig. 1.2.

For instance, in the ASME 2011 Turbo Expo Conference, 24 papers were related to the development and applications of EA-based optimization in the field of thermal turbomachines; some of them are not CFD-based. Most of them are handling a small number of design variables, ($N \leq 15$), and only three (62, 104, 123) solve high-dimensional problems ($N > 30$). In (62), the application of an axisymmetric endwall contour for compressors is investigated. An EA-based optimization of the outer casing and the corresponding blade tip airfoil section of a typical gas turbine high-pressure compressor stage, with a high number of design variables ($N = 38$), is presented. In (123), the high-dimensional ($N = 210$) constrained MOO of a fan stage, using EAs enhanced by metamodels, is presented.

1. Introduction

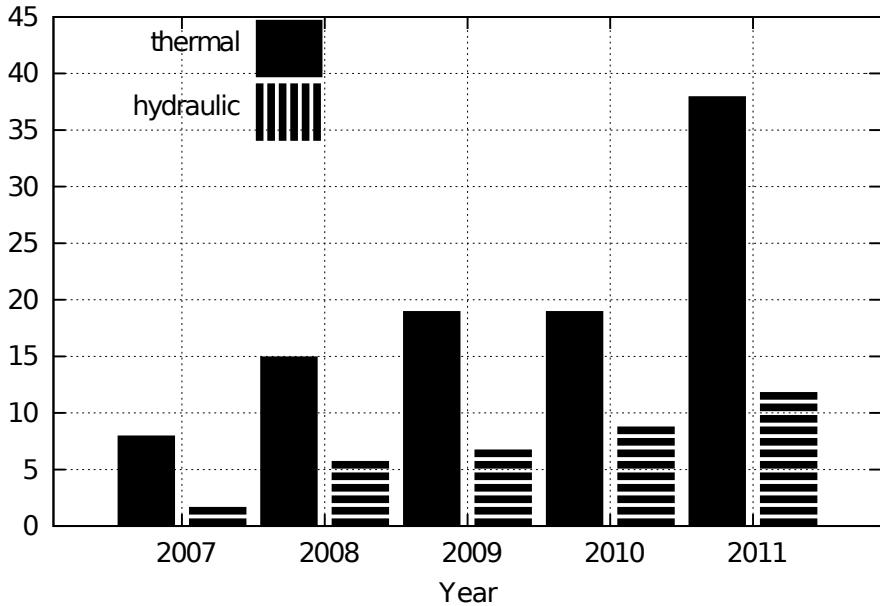


Figure 1.2: Yearly number of recent publications which are relevant to the optimization of thermal and hydraulic turbomachines, in ScienceDirect.

In (104), an EA-based axisymmetric multi-disciplinary optimization approach for compressors is applied to the design of a three-stage booster with $N=53$ design variables. Even though the aforementioned papers use a great number of design variables, they typically start from an almost optimal design and, therefore, are able to significantly restrict the design space per design variable. For instance, (123) starts from a “pre-optimized” design and in (62) the search space of each design variable is of the order of the tip clearance magnitude (very small). Note that the KBD method proposed by this thesis may efficiently deal with both high-dimensional ($N > 300$) problems with extended search space per variable.

In the IAHR 2010 conference, 5 paper on the optimization of hydraulic turbomachines were presented. Three of them, (78, 146, 149) used EAs. In (146), the weekly operation of a multipurpose hydroelectric development, including a pumped storage plant was optimized using an EA. In (149), an optimization problem with two design variables, concerning the design of an axial compressor cascade using a MAEA was presented. In (78), the author of this thesis presented the solution of an optimization problem with 336 design variables, concerning the design of a Francis hydraulic turbine.

Research on the CFD-based optimization in the fields of thermal and hydraulic turbomachines was carried out in the Laboratory of Thermal Turbomachines (LTT/NTUA) and the Laboratory of Hydraulic Machines (LHM/NTUA) of NTUA. Concerning thermal turbomachines, a number of papers regarding both

deterministic and stochastic optimization methods reflect the performed research at LTT/NTUA. Most of them are related to the optimal design of components of thermal turbomachines, such as compressor cascades, using a variety of optimization methods. An indicative subset of them is mentioned below. The coupling of stochastic optimization methods with computational intelligence is presented in (69, 71, 100). There, the use of artificial neural networks as metamodels, in order to assist EAs incorporating costly evaluation tools such as CFD codes, are proposed. In (96), the use of metamodels trained on both responses and gradients is proposed and used in the inverse design of a 3D peripheral cascade. In (101), the hierarchical distributed MAEA is used for the viscous loss minimization of a compressor cascade. In (16), an asynchronous EA is demonstrated on the design of a 2D compressor cascade and in (17) a grid-enabled asynchronous MAEA is applied to the optimization of a 3D annular compressor cascade. In (136), an adjoint-based optimization method is proposed for the total pressure losses minimization in turbomachinery cascades. In (184), the computation of the exact Hessian matrix of the objective function measuring total pressure losses in turbomachinery cascades is presented for use along with Newton methods. Multi-level strategies that combine adjoint-based methods with MAEAs for turbomachines are described in (97).

Concerning the CFD-based optimization in the field of hydraulic turbomachines, the work done at LHM/NTUA is related to: a) the design of optimal components of hydraulic machines, such as runner blades, and b) the optimal design of complete hydroelectric power plants and energy storage plants in combination with other forms of renewable energy generation sources, such as wind energy. Indicatively, in (10), the fast Lagrangian approach is incorporated into an EA to design an optimal Turgo turbine. The optimal sizing of a run-of-river small hydroelectric power plant utilizing an EA is described in (9). In (11, 12), the optimization of pumped-storage systems for wind power plants is presented.

1.2 EAs: Previous work at PCOpt/NTUA

An overview of previous PhD theses on EAs which were carried out by PCOpt/NTUA research group follows.

Giotis' thesis (74), developed a generalized EA able to combine components of the two most widely used evolutionary optimization methods, namely Genetic Algorithms (GAs) and Evolutionary Strategies (ESs). In the same thesis or the corresponding papers (48, 71, 100), the use of artificial neural networks as local metamodels, trained on previously evaluated candidate solutions, was proposed in order to reduce the computational burden. To reduce the wall clock time of the optimization, parallel evaluations of candidate solutions on a number of available

1. Introduction

processors, via the PVM protocol were used. The EASY optimization platform this thesis relies upon originated from Giotis' thesis (74).

Karakasis' thesis (98), further improved the EA efficiency by focusing on the optimal use of metamodels in MOO problems. One of the most important outcomes of this thesis was that the MAEA became equally efficient in both SOO and MOO, by overcoming problems related to: (a) the fact that, in MOO, the previously computed individuals, by the EA are spread along a front in the design space rather than clustered around the sought optimal solution and (b) the presence of outliers that require special treatment. Distributed EAs (DEAs) were programmed and validated, in order to increase the parallel efficiency of EAs and avoid stagnation during the early generations. Furthermore, the notion of Hierarchical EAs (HEAs) was introduced and this was later refined in (92). In (98), the use of a DEA on each level of the HEA associated with a different evaluation model was proposed.

In Kampolis' thesis (92), the HEAs were upgraded by introducing hierarchical schemes other than those based on the combined use of low- and high-fidelity evaluation software. The latter made use of different parameterization schemes (coarse and fine) or different search methods allowing, thus, the combination of EAs with deterministic optimization methods. Finally, new artificial neural networks trained on both the objective functions values and their gradient were proposed, see also (96).

Asouti's thesis (15), proposed a different way to increase the parallel efficiency of EAs or MAEAs. This was achieved by introducing the so-called Asynchronous EAs (AEAs) or Asynchronous MAEAs (AMAEAs). By utilizing a number of strongly interconnected demes, according to a newly proposed topology, the asynchronous variants may circumvent the “end of generation” synchronization barrier and, thus, uninterruptibly use all the available processors. The proposed AEAs and AMAEs are appropriate for heterogeneous multi-processor platforms.

Georgopoulou's thesis (63), proposed the so-called Metamodel-Assisted Memetic Algorithm (MAMA). Memetic Algorithms (MAs) are hybrid methods that combine stochastic and deterministic search methods. The proposed MAMA combines the advantages of both MAEAs and MAs.

In a sixth PhD (115), Kontoleontos also used the EASY platform to optimize energy production systems, such as geothermal power plants and ground source heat pump systems.

1.3 Thesis Outline

A short overview of the chapters of this PhD thesis follows:

Chapter 2 presents the pre-existing algorithmic basis of this PhD. This in-

1.3 Thesis Outline

cludes the generalized EA with its evolution operators, the MAEA and the HEA.

Chapter 3 is concerned with the first innovative method proposed in this PhD thesis, namely the Knowledge-Based Design (KBD) one. In the first part of this chapter, a short overview of KBS is presented. Then, the proposed KBD method is described in detail and used to perform the design of a 2D compressor cascade, in order to demonstrate its merits.

Chapter 4 is dealing with EAs and MAEAs suitable to solve ill-posed optimization problems. This chapter starts by describing the features of ill-posed optimization problems. Then, the drop in EA efficiency when used to solve ill-posed problems is investigated. An innovative way to find new directions in the design space that, if used to redefine the design variables, would result in a better-posed problem that can be solved using EAs and MAEAs at lower computational cost is proposed. New evolution operators, the so-called PCA-driven ones, are devised, in order to recover the aforementioned EA efficiency drop. Furthermore, the use of the importance information, resulting from the PCA is used to enhance metamodel's efficiency. Finally, the gain achieved using the proposed methods is quantified through a 2D compressor cascade design-optimization case.

In Chapter 5, the methods proposed in Chapters 3 and 4 are used to solve large-scale industrial problems in the field of hydraulic turbomachines. The parameterization, grid generation and CFD tools used in this studies are presented. Quality metrics measuring each hydraulic turbine quality with respect to cavitation, blade loading and draft-tube coupling are introduced. Then, the optimization of a Francis turbine, performed in the context of a modernization/rehabilitation project, is used to demonstrate the merits of the KBD method, by comparing it with a conventional EA. Also, the optimization of a new type of hydraulic turbine, the so-called Hydromatrix®, is carried out in order to demonstrate the gain expected from MAEAs using the proposed PCA-driven evolution operators.

Chapter 6 is dedicated to the design-optimization of the compressor cascade installed at LTT/NTUA. This case is used to investigate the combined effects of using the PCA-driven evolution operators and the PCA-assisted metamodels, proposed in Chapter 4.

Finally, Chapter 7 summarizes the conclusions drawn in this PhD thesis.

1. Introduction

2

Evolutionary Algorithms (EAs)

2.1 Introduction - Overview of EAs

Several stochastic optimization methods have been inspired by and/or based on Darwin’s theory of evolution on the origin of species (33). Hereafter, the aforementioned methods along with their variants (there are plenty of them) will be referred to as Evolutionary Algorithms (EAs) (126). The increasing power of modern multi-processor computational platforms and their availability at relatively low cost, combined with the inherent parallelization of population-based algorithms, suggest that EAs, combined with existing analysis tools, can be used as efficient industrial design tools. In our case, the analysis or problem-specific solution software is a Computational Fluid Dynamics (CFD) code. The cost of running the CFD code, quite often more than once per candidate solution (such as in case of a multi-point design) times the number of required trials determines the overall computational cost of the EA-based optimization.

An EA is a generation-based, stochastic optimization method handling populations of individuals evolving from generation to generation. Each individual \vec{x} represents a potential solution to the optimization problem in hand and must be evaluated to obtain a measure of its “fitness” or “cost”, according to user-defined objective functions. In each generation, the EA selects the most fit among the previous generation members (the so-called “parents”) and evolves them by applying evolution operators (recombination, mutation, etc.). The new population (“offspring”) is expected to fit better to the environment determined by the selected objective function(s).

The first attempts to use EAs as problem solving techniques are dated back to mid-50’s and can be found in separate works by Friedberg, Bremermann and Box. Friedberg (57, 58) presented an EA capable of creating a program to perform a given task; though quite premature by that time, this was in fact an ancestor of evolutionary programming (see below). During the same period of

2. Evolutionary Algorithms (EAs)

time, Bremermann (23) was among the first to apply EAs to numerical (linear and convex) optimization problems including the solution of systems of nonlinear equations. Box proposed the use of EAs as a method for improving industrial processes and increasing productivity (21, 22). By that time, these first attempts were treated with considerable skepticism. Soon after, several methods which initiated the three broad classes of EAs, namely evolutionary programming (EP), evolutionary strategies (ES) and genetic algorithms (GA), were brought to light.

EP was proposed by L. J. Fogel by mid-60's (51, 52, 53). EP was developed by considering machine learning tasks by means of finite-state machines (FSM¹). The optimization problem was initially defined as evolving an algorithm (program) for predicting arbitrary time series. In EP, (53), each offspring is created by randomly mutating each parent. The offspring with the best value is, then, retained to become a parent in the next generation. EP was successfully applied to problems in prediction identification, automatic control and pattern recognition. Until mid-80's, EP was confined to the FSM representation. In 1986 EP was extended to alternative representations including ordered lists for the travelling salesman problem and real-valued vectors for continuous function minimization. Later on, a self-adaptive EP (54) was proposed by including mutation variance in the evolution.

The first ES was proposed in 1965 by Rechenberg (151) using discrete, binomially distributed mutations (centered at the parent's position) and just a single parent and a single offspring per generation. Later on, ES was further developed both by Rechenberg (152) and Schwefel (164), by introducing recombination and adaptive mutation (152). Using real-valued representation of the optimization variables, mutation is performed by adding a normally distributed random value to them. The introduction of recombination as an additional evolution operator for creating offspring, being impossible with just a single parent, led to the multi-membered ES. This is usually referred to as a (μ, λ) ES, i.e. an ES with μ parents and λ offspring. One of the most successful variants of ES is the covariance matrix adaptation ES (CMA-ES) (79). In CMA-ES, the covariance matrix of the multivariate normal distribution, from which the candidate solutions are sampled, is updated during the evolution. CMA-ES was found to be particularly useful in solving non-separable optimization problems. (80, 81).

In 1962, GAs were proposed by Holland (84, 85) as an evolution emulating tool for aiding the understanding of the underlying principles of adaptive systems². In comparison to other contemporary researchers in the field of EAs, Holland

¹A finite-state machine (FSM) is a mathematical model used to design computer programs and digital logic circuits. FSM is an abstract machine that can be in one among a finite number of states.

²Systems that are capable to undergo self-modifications in response to their interactions with the environment which they must function in.

2.1 Introduction - Overview of EAs

focused differently and laid emphasis on the algorithmic nature of GAs in the sense that the reproduction and inheritance mechanisms were based on operators, such as mutation, crossover and inversion, which are well known from genetics¹. In addition, the representation of the evolving objects was made by using binary strings in direct analogy to the genetics genome. The binary string representation was one of the distinctions between ES and GA, considered as important at least in the early stages of their lives.

Genetic Programming (GP) was proposed by Crammer in mid-80's (32) and was, then, improved by Koza (117) aiming at the automated design of computer programs with the ability to perform a given computational task. In GP, individuals (computer programs) are represented as tree structures² where each node is given an operator function and each terminal node an operand. Representing programs in this way offers easy evaluation and makes the application of the evolution operators possible, in the sense that recombination can take place by exchanging nodes between the parents whereas mutation by replacing nodes at random.

In PCOpt/NTUA, GA and ES are both represented by a generalised $(\mu, \lambda)EA$ (74, 92, 98), as described in section 2.3. The $(\mu, \lambda)EA$ main characteristics are the population-based evolution and the fact that the inheritance of candidate solution features is based on both probabilistic criteria and decisions made upon a fitness or cost function, which quantifies the ability of each individual to survive in the user-defined environment. The evolution, from each generation to the next, fig. 2.1, is carried out through the so-called evolution operators (parent selection, recombination, elitism and mutation). The generalised $(\mu, \lambda)EA$, which is used and enhanced in this PhD thesis, can be used as either a conventional ES or a conventional GA, by carefully adjusting the algorithmic settings and the optimization variable representation (binary or real coding).

The three main advantages of EAs are their ability to (a) avoid being trapped into local optima and, thus, locate the global optimum, (b) compute Pareto fronts of optimal solutions in MOO problems, with a single run and (c) accommodate any ready-to-use analysis software without requiring access to its source code. The only prerequisite for carrying out an EA-based optimization is the availability of an appropriate evaluation software (even a commercial one, considered as a black-box tool within the optimization loop), objective functions and well defined design variables along with lower and upper bounds for each of them.

¹Genetics is a branch of biology, the science of genes, heredity and the variation in living organisms.

²A tree structure can represent graphically the hierarchical nature of an object. Tree structures start from the so-called "root" node, which is the highest in hierarchy. Lines connecting elements, to be referred to a nodes, are called "branches". The lowest in hierarchy nodes are called "end-nodes" or "leaves".

2. Evolutionary Algorithms (EAs)

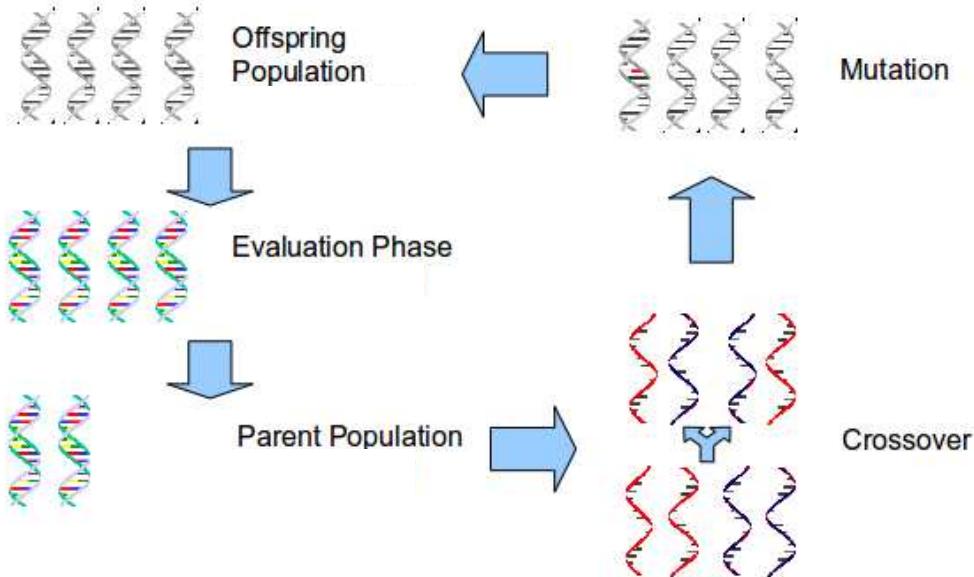


Figure 2.1: Schematic representation of an EA. Each population is derived by the previous one via the application of evolution operators such as parent selection, crossover and mutation. Elitism is not shown here.

However, the need to evaluate all individuals in each generation, so as to assign fitness or cost values to all of them, is the weak point of EAs, particularly if the evaluation of each candidate solution is computationally demanding. Optimization in aerodynamics or hydrodynamics which relies on expensive CFD software is a typical example. This increases noticeably the optimization turn-around time and, often, makes the whole process non-affordable for industrial use. To overcome this weakness, several techniques have been proposed, (69, 90). These are classified in those reducing the optimization turn-around time by performing concurrent evaluations of the population members on a multi-processor platform and those reducing the number of evaluations needed to reach the optimal solution(s). These two techniques can certainly be combined. A first overview of these techniques is given below.

A Parallel Evolutionary Algorithm (PEA) (15, 74, 92, 98) is an EA adapted to take advantage of the availability of multi-processor systems, for reducing the optimization turn-around time. PEAs can be classified into single- and multi-population EAs. In a single-population or panmictic EA, each population member can, potentially, mate (be recombined) with any other. Standard way of parallelization is the concurrent evaluation scheme, with centralized selection and evolution (the so-called “master-slave” model). A multi-population EA handles partitioned individual subsets, according to a topology which often maps directly

2.1 Introduction - Overview of EAs

onto the parallel platform and employs decentralized evolution schemes; these can be further classified into distributed (83, 94, 100) and cellular EAs (7, 132), depending on the subpopulations' structure and granularity.

All the aforementioned PEAs refer to synchronous EAs, in which the use of the multi-processor platform is restricted to the concurrent evaluation of candidate solutions, without altering the, generation-by-generation, sequential nature of evolution in EAs. This creates a synchronization barrier at the end of each generation; at this point, a number of processors may remain idle while waiting for the remaining generation members to be evaluated. In order to minimize (practically eliminate) the idle time of processors, asynchronous EA (AEA) (8, 16) have been developed. The AEA proposed by PCOpt/NTUA overcomes the sequential nature of evolution by applying the evolution operators to a number of strongly interacting (overlapping) demes which may optimally use all the available processors.

A metamodel-assisted EA (MAEA) employs low-cost surrogate evaluation models, i.e. the so-called “metamodels”, as often as possible, during the optimization. This decreases substantially the number of calls to the computationally expensive, problem-specific evaluation code (such as the CFD software). Polynomial regression, artificial neural networks, Gaussian processes etc. have all been used as metamodels. Schemes based on different interactions between the metamodel and the problem-specific evaluation tool can be found in the literature (47, 69, 90, 103). MAEA implementations can be classified in two basic categories, depending on whether the metamodel(s) is/are trained separately from (off-line) or during (on-line) the evolution. Additionally, global (i.e. a single metamodel for the entire search space) or local (i.e. each of which being valid over a different part of the search space) metamodels can be used.

Additional reduction in the optimization turn-around time can be achieved by employing hierarchical EAs (HEAs) or MAEAs (HMAEAs)(83, 92, 94, 98, 101, 122). These are built based on a small number of interconnected levels. On each level, different search methods, different evaluation software and/or different sets of design variables can be employed. One- or two-way inter-level communication schemes can be used. On levels employing EA-based search, a PEA or MAEA or both can optionally be used. Multi-level optimization algorithms can, generally, be classified as follows (73):

(a) *Multi-level Evaluation*, where each level is associated with a different evaluation software. The lower levels are responsible for detecting promising solutions at low CPU cost, through low-cost problem-specific evaluation models, and delivering them to the higher level(s). There, evaluation models of higher fidelity and CPU cost are employed and the immigrated solutions are further refined.

(b) *Multi-level Search*, where each level is associated with a different search technique. Stochastic methods, such as EAs, are preferably used on the lower

2. Evolutionary Algorithms (EAs)

level(s) for the exploration of the design space, leaving the refinement of promising solutions, often via gradient-based methods, on the higher level(s). Other combinations of search tools are also possible. Memetic algorithms (64, 65, 116, 118, 133, 134) is a class of multi-level search that combines global and local search methods, where the latter aims at improving the quality of promising solutions.

(c) *Multi-level Parameterization*, where each level is associated with a different set of design variables. On the lowest level, a subproblem with just a few design variables is solved. On the higher level(s), the problem dimension increases. The most detailed parameterization is used on the highest level.

2.2 Optimization Problems - Definitions

Any optimization problem with M objectives (cast in vector \vec{F}) and K constraints (cast in vector \vec{C}) can be formulated as:

$$\begin{aligned} \min \vec{F}(\vec{x}) &= (f_1(\vec{x}), f_2(\vec{x}), \dots, f_M(\vec{x})) \in \Re^M \\ \text{subject to } c_k(\vec{x}) &\leq d_k, \quad k = 1, K \end{aligned} \tag{2.1}$$

where $\vec{x} \in X \subseteq \Re^N$ is the design vector and X the design space. If equality constraints $c^*(x) = d^*$ are to be imposed, these can be transformed into inequality ones $c(x) = \|c^*(x) - d^*\| \leq d$, where $d \in \Re$ is an infinitessimally small number. If $M > 1$, problem 2.1 represents a multi-objective optimization (MOO) problem; in such a case, the notion of Pareto dominance (186) is used to associate a scalar cost value to each individual, based on which an algorithm solving single-objective optimization (SOO) problems can be employed. In MOO problems, a single run of an EA is capable of delivering a Pareto front of non-dominated individuals, rather than a single “optimal” solution, standing for an offset among the various objective functions.

2.2.1 Multi-Objective Optimization and EAs

In contrast to SOO where the scalar cost function, which determines the survival of candidate solutions from generation to generation, is derived directly from the objective function, a solver for MOO problems handles a vector of objectives. Therefore, in order to rely on the EA built for solving SOO problems, a technique to transform this vector to an appropriate scalar cost value is needed. In the literature, several techniques for handling this problem exist (27, 30, 129). The most commonly used among them concatenate the M objective functions into a scalar cost function, either by associating weights to each one of them or by accounting for the distance between the candidate solution and a user-defined ideal point in

2.2 Optimization Problems - Definitions

the objective space or, even, by using ranking techniques based on the notion of Pareto dominance (see below). In this PhD thesis, all MOO problems are handled using techniques relying on Pareto dominance criteria. Different implementations of Pareto dominance based techniques exist in the literature; among them, the most widely used ones are MOGA¹ (55), NPGA² (86), NSGA³ (170), NSGA2 (36), SPEA⁴(188), SPEA2 (187), PAES⁵ (107) and PESA⁶ (31). The definition of Pareto dominance and Pareto optimality in minimization problems, which the aforementioned techniques rely on, follow:

Pareto Dominance: Solution \vec{x}_1 dominates solution \vec{x}_2 ($\vec{x}_1 \prec \vec{x}_2$) if and only if $\vec{F}(\vec{x}_1)$ is partially less than $\vec{F}(\vec{x}_2)$, i.e.

$$\vec{x}_1 \prec \vec{x}_2 \Leftrightarrow (\forall i \in [1, M] : f_i(\vec{x}_1) \leq f_i(\vec{x}_2)) \wedge (\exists_i : f_i(\vec{x}_1) < f_i(\vec{x}_2)) \quad (2.2)$$

Pareto Optimality: Solution $\vec{x}_1 \in X$ is a Pareto optimal solution with respect to X if and only if there is no $\vec{x} \in X$ that dominates \vec{x}_1 , or

$$\nexists \vec{x} : \vec{x} \prec \vec{x}_1, \quad \vec{x}, \vec{x}_1 \in X \subseteq \Re^N \quad (2.3)$$

In fig. 2.2, a schematic representation of the Pareto optimality is shown. A minimization problem with two objective functions, f_1 and f_2 , is considered. Black circles denote the front of non-dominated individuals whereas the empty circles denote dominated individuals. In the sake of clarity, the part of the objective space which is dominated by \vec{x}_1 is highlighted in grey. Individuals located in the grey area are dominated, at least, by \vec{x}_1 .

A detailed presentation of three techniques (SPEA, SPEA2 and NSGA2) used in this thesis follows in section 2.3.2.

2.2.2 Constrained Optimization and EAs

In engineering, almost all optimization problems are subject to constraints which split the design space into feasible and infeasible regions. EAs may handle constraints through (a) the use of penalty functions (34, 131), by assigning greater

¹Multi-objective Genetic Algorithm.

²Niched Pareto Genetic Algorithm.

³Non-Dominated Sorting Genetic Algorithm.

⁴Strength Pareto Evolutionary Algorithm.

⁵Pareto Archived Evolution Strategy.

⁶Pareto Envelope-based Selection Algorithm.

2. Evolutionary Algorithms (EAs)

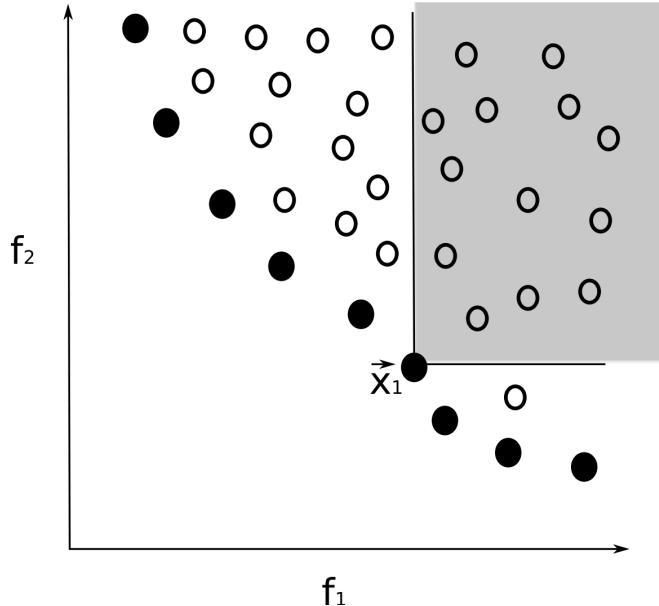


Figure 2.2: Schematic representation of Pareto dominance in a two-objective minimization problem ($\min(f_1)$ and $\min(f_2)$). \vec{x}_1 is a Pareto optimal solution since this is not dominated by any other solution. \vec{x}_1 dominates all individuals located in the grey box. All Pareto optimal solutions are marked with black circles.

chances to survive to individuals satisfying the constraints (147), (b) the conversion of constraints into objectives (171, 172) and/or (c) the use of correction operators (128). A detailed survey on the subject can be found in (29, 127). In the present thesis, penalty functions are used as presented in section 2.3.3.

2.3 The Evolutionary Algorithm SYstem (EASY)

EASY is a generic optimization platform which implements the $(\mu, \lambda)EA$, (70, 74, 92, 98). EASY has been developed by PCOpt/NTUA and was used as the algorithmic basis for all techniques proposed in this thesis. EASY supports MAEAs using on- or off-line trained metamodels, various forms of multi-level optimization as described in section 2.1, distributed and asynchronous search; it is also Grid/Cluster-computing enabled, based on the DRMAA library, (121). A detailed analysis of the implemented $(\mu, \lambda)EA$, considered as the background optimization method in this thesis, follows.

2.3 The Evolutionary Algorithm SYstem (EASY)

2.3.1 The (μ, λ) EA

Each generation, denoted by superscript g , of the (μ, λ) EA is associated with three dynamically updated populations: the offspring P_λ^g population with λ offspring, the parent P_μ^g population containing μ parents and the elite P_e^g population with e elites. P_e^g contains all or some of the currently best individuals. Depending on the design variables coding (binary, binary Gray or real), appropriate evolution operators are applied.

The background (μ, λ) EA in use, (74), comprises the following steps:

Step 1: (Initialization) $g = 0$, $P_e^{g-1} = \emptyset$ and $P_\mu^g = \emptyset$. All P_λ^g members are initialized using a pseudo-random number generator, in accordance to the user-defined lower and upper bounds of each design variable. A number of user-defined individuals, such as existing sub-optimal solutions to the same problem or optimal solutions to similar problems, can optionally be included into the initial population.

Step 2: (Evaluation) All individuals in P_λ^g are evaluated using the problem-specific evaluation model. In all applications this thesis is dealing with, this is a CFD code. Later, when dealing with MAEAs, this will also be referred to as “exact evaluation model” to clearly distinguish it from evaluations (pre-evaluations) on cheaper surrogate models. The evaluation step is, always, considered to be time-consuming and yields vectors $\vec{F}(\vec{x}) \in \Re^M$ for each $\vec{x} \in P_\lambda^g$.

Step 3: (Cost assignment) For each $\vec{x} \in P_\lambda^g \cup P_\mu^g \cup P_e^{g-1}$, a scalar cost $\Phi(\vec{x})$ value is assigned based on $\vec{F}(\vec{x})$. In SOO, $\Phi(\vec{x}) \equiv F(\vec{x})$. In MOO, the techniques presented in section 2.3.2 are employed.

Step 4: (Elite selection) The e^* non-dominated individuals (in a MOO problem) or the best one (SOO) in $P_\lambda^g \cup P_e^{g-1}$ are selected to enter P_e^g . If $e^* > e$, a thinning operator (74) can be applied to remove the $e^* - e$ excess individuals.

Step 5: (Elitism) A few elite individuals, selected at random from P_e^g , replace the worst members of P_λ^g .

Step 6: (Parent selection) The new parent population P_μ^g is selected from P_μ^{g-1} and P_λ^g by taking into account the scalar cost values Φ computed in step 3 and the maximum allowed age κ (measured in generations) of each individual. Symbolically, $P_\mu^g = S(P_\mu^{g-1}, P_\lambda^g, \kappa)$

Step 7: (Recombination and mutation) The next generation of the offspring population P_λ^{g+1} is derived from P_μ^g by applying the recombination

2. Evolutionary Algorithms (EAs)

(or crossover) and mutation operators. Recombination $\mathcal{R}()$ is the process of combining the genotypes of ρ parents to create an offspring (section 2.3.1.1). The recombination operator is used λ times, with different sets of ρ parents, randomly selected from P_μ^g , to create λ new offspring. Next step is the application of the mutation operator. Mutation $\mathcal{M}()$ is a process which, with a small probability, randomly alters parts of the individual genotype. Symbolically, $P_\lambda^{g+1} = \mathcal{M}(\mathcal{R}(P_\mu^g))$ (section 2.3.1.1).

Step 8: (Stopping criterion) Unless any of the stopping criteria is met, return to *Step2*.

The basic nomenclature of the (μ, λ) EA is given in table 2.1.

Number of design variables	N
Number of objectives	M
Number of constraints	K
Candidate solution, design vector	$\vec{x} = (x_1, \dots, x_N)$
Objective function	$f_i(\vec{x})$
Vector of objective values (if M>1)	$\vec{F} = (f_1(\vec{x}), \dots, f_M(\vec{x}))$
Constraint function	$c_i(\vec{x})$
Vector of constraint values	$\vec{C} = (c_1(\vec{x}), \dots, c_K(\vec{x}))$
Scalar cost value ($\Phi = f$ for M=1)	Φ
Number of offspring	λ
Number of parents	μ
Number of elites	e
Maximum allowed age (in number of generations)	κ
Number of parents per offspring	ρ

Table 2.1: Nomenclature of the (μ, λ) EA.

2.3.1.1 Evolution Operators

The evolution operators employed in EASY are the parent selection, recombination, mutation and elitism. These are further discussed below:

Parent Selection: The parent selection operator selects the members of P_μ^g from a set of already evaluated individuals. In EASY, based on the sixth step of the algorithm presented in section 2.3.1, this set is the union of P_μ^{g-1} and P_λ^g . The most common parent selection techniques (19) are:

2.3 The Evolutionary Algorithm SYstem (EASY)

- a) **Proportional Selection:** Each member of $P_\mu^{g-1} \cup P_\lambda^g$ is given a probability to become a parent, which is inversely proportional to its cost value Φ . Recall that a minimization problem is to be solved. The proportional selection is, practically, implemented via a roulette wheel. On the roulette wheel, each and every set member is given a slot, the angular size of which is proportional to its probability to become a parent. The roulette wheel is used μ times so as to create μ parents.
- b) **Linear Ranking:** In linear ranking, the $P_\mu^{g-1} \cup P_\lambda^g$ members are sorted according to their Φ values. The probability of each one of them to become a parent depends linearly on its position in the corresponding list and not on the Φ values themselves.
- c) **Probabilistic Tournament Selection:** In probabilistic tournament selection, (77), a number of individuals are selected at random from the $P_\mu^{g-1} \cup P_\lambda^g$ set and, with a user-defined (typically high) probability, the best individual from this group is selected to become a parent. Otherwise, a randomly chosen one among the remaining individuals becomes parent. This is repeated μ times. In tournament selection, the most important user-defined parameters are the tournament size, i.e. how many individuals participate in each tournament and the probability of the best among them to be selected as parent. Typical values are: participation of 2 or 3 individuals in each tournament and 80%-95% probability to select the best among them.

Recombination: Since the first appearance of EAs, numerous discussions about the advantages of employing the recombination operator can be found (159, 160, 169). In general, the purpose of recombination is to increase the probability of an offspring to become fitter than its parent(s). Recombination schemes can be classified depending on the design variables' coding.

In EAs based on **binary coding**, the recombination operator undertakes the exchange of pieces of binary strings encoding the design vector between the parents, in order to create an offspring. In EASY, the following recombination operators, which are appropriate for use with binary coding, exist:

- a) **One-Point Recombination:** In the one-point binary recombination, with ρ parents per offspring, the binary string is divided into $\rho-1$ parts of equal length in binary digits. Then, ρ parents are selected at random from the P_μ^g set and, using them, $(\rho-1)$ pairs of parents are formed; all of them include the first parent. If the first parent is denoted by 1, the $(\rho-1)$ pairs are: $(1, 2), (1, 3), \dots, (1, \rho)$. For each one of the $(\rho-1)$ pairs, a random integer X determines the corresponding crossover point. Each offspring is formed by combining the left part of the first parental string and the right part of

2. Evolutionary Algorithms (EAs)

the other parent, according to the aforementioned pairings. A three-parent example ($\rho = 3$) is presented in fig. 2.3.

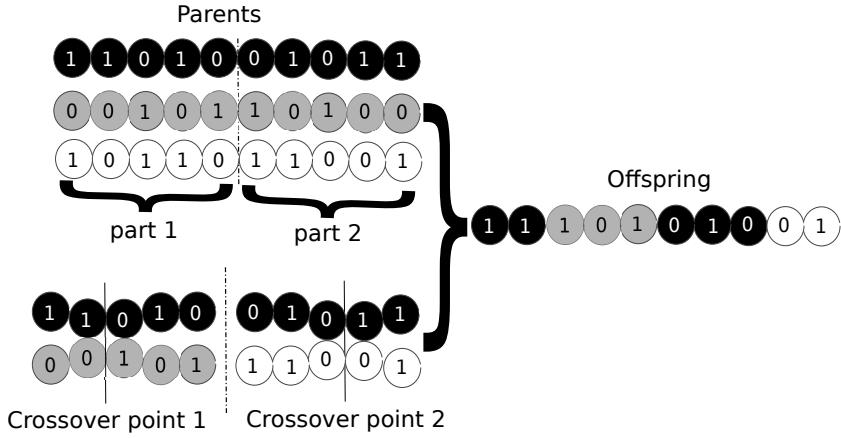


Figure 2.3: One-point binary coding recombination with $\rho = 3$. In this case, the chromosome is divided in two parts ($\rho - 1 = 2$) and, then a vertical cut per part is created at random.

b) Two-Point Recombination: This scheme is similar to its one-point counterpart, the only difference being that two crossover points X_1 and X_2 are used. A two-parent example is illustrated in fig. 2.4.

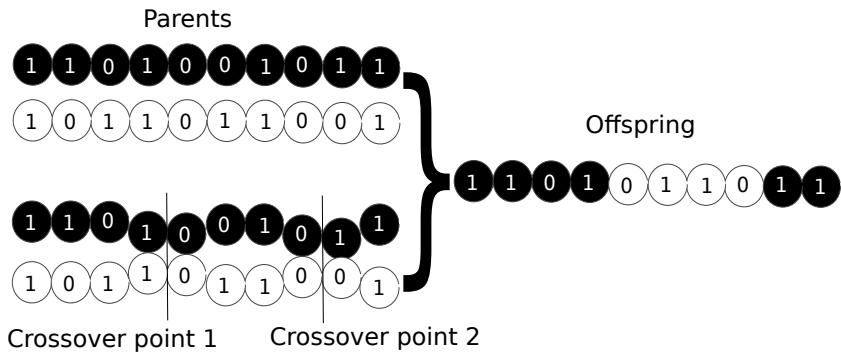


Figure 2.4: Two-point binary coding recombination with $\rho = 2$. In this simple case, the chromosome is handled as a whole ($\rho - 1 = 1$) and two vertical cuts are created at random.

c) One- or Two-Point Recombination per Design Variable: Here, the aforementioned one- and two-point recombination schemes are separately applied to the parts of the binary string that correspond to each design variable.

2.3 The Evolutionary Algorithm SYstem (EASY)

Using **real coding**, the recombination operator applies directly to the real-valued design variables. In EASY, this is carried out as follows:

- a) One- or Two-Point Recombination:** The one- and two-point recombination schemes, as described for binary coding, can be adapted to real coding by exchanging pieces of the design vector \vec{x} instead of pieces of binary strings. The randomly selected crossover variable is the only one to be affected by both parents according to a randomly generated weight r . A one-point, two-parent example is presented in fig. 2.5.

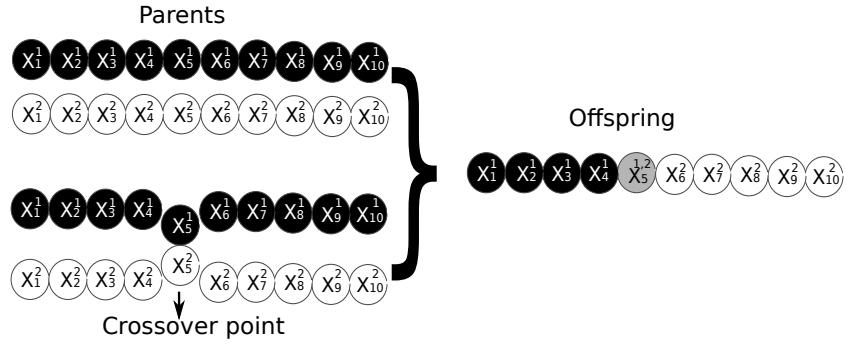


Figure 2.5: One-point real coding recombination with $\rho = 2$. The superscript and subscript denote the parent and the design variable respectively. The crossover variable (e.g. 5) is selected at random. In the offspring, this variable is given by $x_5^{1,2} = x_5^1 + r(x_5^2 - x_5^1)$, where r is a random number, uniformly distributed in $[0, 1]$ ($r \in U(0, 1)$).

- b) Discrete Recombination:** In discrete recombination with ρ parents, each real-valued design variable in the offspring has 50% probability to be copied either from the first parent \vec{x}^1 or any other \vec{x}^{random} , randomly chosen among the $\rho-1$ remaining ones. A three-parent example is presented in fig. 2.6.

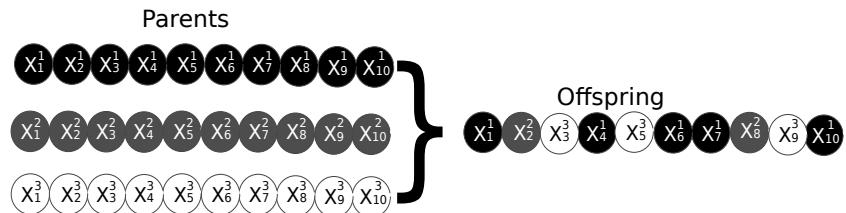


Figure 2.6: Discrete real coding recombination with $\rho = 3$. \vec{x}^1 is the first parent.

- c) Intermediate Recombination:** In intermediate recombination, each component of offspring \vec{x} is a linear combination of the first parent \vec{x}^1 and

2. Evolutionary Algorithms (EAs)

\vec{x}^{random} , which may be any other parent randomly chosen among the $\rho - 1$ remaining ones. Thus,

$$\vec{x} = \vec{x}^1 + r(\vec{x}^{random} - \vec{x}^1), \quad r \in U[0, 1]$$

where $r \in U(0, 1)$ means that r is a random number, uniformly distributed in $[0, 1]$. A different random number is generated for each design variable.

d) Simulated Binary Crossover: The so-called simulated binary crossover (SBX) (35) aims at recreating the one-point binary recombination by replicating its (a) average property¹ and (b) spread factor property² (35). The offspring \vec{x} is generated based on the formulas

$$\vec{x} = \begin{cases} \vec{x} - \frac{\beta}{2}(\vec{x}^{random} - \vec{x}^1) & , r < 0.5 \\ \vec{x} + \frac{\beta}{2}(\vec{x}^{random} - \vec{x}^1) & , r \geq 0.5 \end{cases} \quad (2.4)$$

where $r \in U[0, 1]$, \vec{x}^1 the first parent, \vec{x}^{random} is randomly selected among the $\rho - 1$ remaining parents, \vec{x} is the middle point between \vec{x}^1 and \vec{x}^{random} and the spread factor β is defined by

$$\beta = \begin{cases} (2r)^n & , r \leq 0.5 \\ \left(\frac{1}{2r}\right)^{n+2} & , r > 0.5 \end{cases} \quad (2.5)$$

Given that $r \in U[0, 1]$, the probability distribution of β can be plotted for various values of n based on eq. 2.5 (fig. 2.7, left). Based on the β probability distribution and eq. 2.4, the probability distribution of offspring appearance on the design space as a function of n can be plotted (fig. 2.7, right for a 1D problem and fig. 2.8 for a 2D problem). In general, higher n values increase the probability of $\beta \approx 1$, which in turn increases the probability of creating near-parent offspring (spread factor property). This, together with the symmetrical offspring probability distribution with respect to \vec{x} (average property), show that SBX may in fact simulate the one-point binary crossover.

Mutation: In EAs, the role of mutation is to introduce and preserve diversity in the population. Thus, mutation prevents the population from becoming

¹The average of the decoded variable values is the same before and after the crossover operation.

²The spread factor (β) is defined as the ratio of the spread of offspring to that of parents. Contracting, expanding or stationary recombination correspond to $\beta < 1$, $\beta > 1$ or $\beta = 1$ respectively. The spread factor property states that the occurrence of spread factor $\beta \approx 1$ is more likely than any other β value.

2.3 The Evolutionary Algorithm SYstem (EASY)

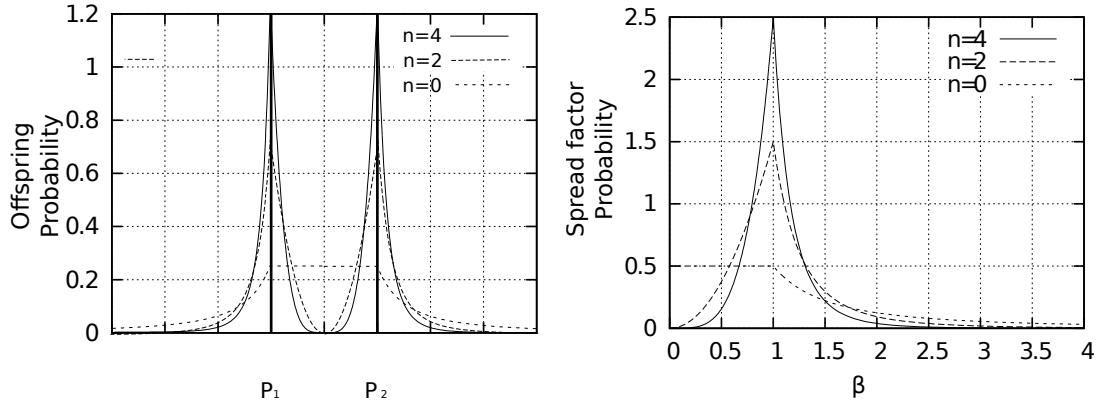


Figure 2.7: The offspring probability distribution of a 1D problem using SBX, with various values of n , is plotted as a function of the parents P_1 and P_2 (left). The corresponding probability distributions of β are plotted on the right. Increasing n leads to higher probability of $\beta \approx 1$.

prematurely homogenized, in which case slow or stagnated evolution may occur. Mutation is applied to each offspring generated using the recombination operator, subject to a small mutation probability P_m . From the implementation viewpoint, mutation depends on whether binary or real coding is used.

In EAs based on **binary coding**, mutation is applied to each and every bit of the binary string. Each bit can be inverted (flip¹), with a probability P_m .

In **real coded EAs**, mutation is applied to each real-valued component of \vec{x} ,

$$\vec{x}_m = \begin{cases} \vec{x} & , P_m > r \\ M(\vec{x}, D) & , P_m \leq r \end{cases} \quad (2.6)$$

where $r \in U[0, 1]$ seeded separately for each member of \vec{x} and

$$M(\vec{x}, D) = \begin{cases} \vec{x} + D(g, \vec{U} - \vec{x}) & , r_1 > 0.5 \\ \vec{x} - D(g, \vec{x} - \vec{L}) & , r_1 \leq 0.5 \end{cases} \quad (2.7)$$

where $r_1 \in U[0, 1]$, \vec{U} and \vec{L} are the vectors of upper and lower bounds of \vec{x} respectively and

$$D(g, y) = yr_2(1 - g/g_{max})^{0.2} \quad (2.8)$$

¹Bit flip is a state switch, from 0 to 1, or vice-versa.

2. Evolutionary Algorithms (EAs)

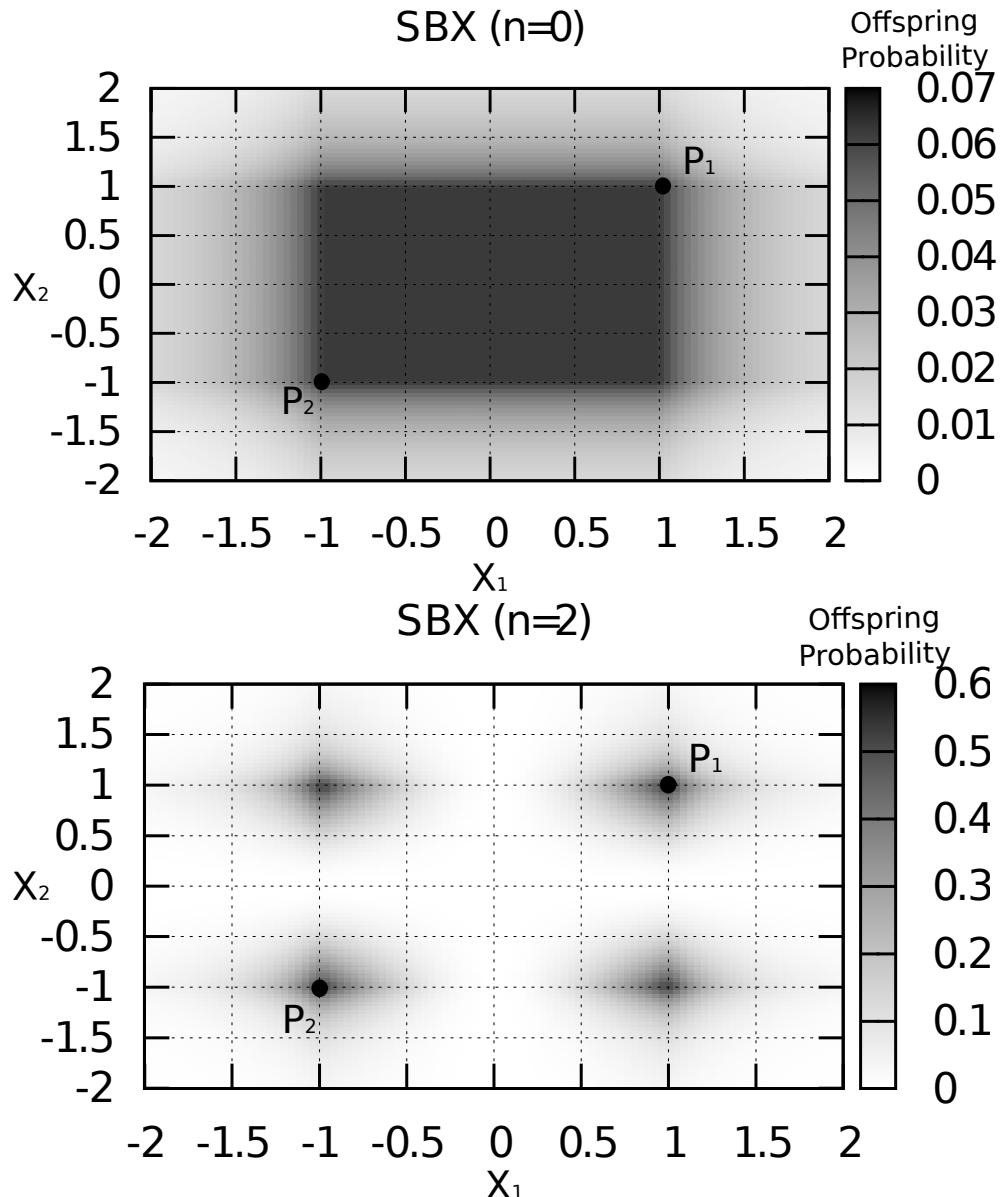


Figure 2.8: The offspring probability distribution in a 2D optimization problem using SBX recombination for $n = 0$ (top) and $n = 2$ (bottom) is presented. The recombination uses two parents, namely $P_1(1, 1)$ and $P_2(-1, -1)$.

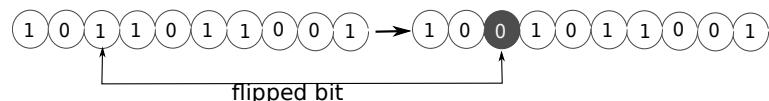


Figure 2.9: Binary coding mutation.

2.3 The Evolutionary Algorithm SYstem (EASY)

where g_{max} the number of maximum generations and $r_2 \in U[0, 1]$. Alternatively, the termination criterion can be based on the number of evaluations, so g/g_{max} must be replaced with Ev/Ev_{max} , where Ev is the evaluation counter and Ev_{max} the maximum allowed number of evaluations.

Elitism: In EAs, the purpose of elitism is to guarantee a monotonously improving course of evolution (19). In EASY, a separate population P_e^g is maintained and updated accordingly at the end of each generation (step 4 of the (μ, λ) EA, section 2.3.1). Elitism implies that a user-specified number of elite individuals replace the worst members of the offspring population P_λ^g , prior to the application of the parent selection operator.

2.3.2 Solving MOO Problems with EASY

As mentioned in section 2.2.1, in MOO problems, a technique that transforms \vec{F} into a scalar cost function Φ is required. Symbolically,

$$\Phi(\vec{x}) = \Phi(\vec{F}(\vec{x})) : \Re^M \rightarrow \Re^1 \quad (2.9)$$

The SPEA, SPEA2 and NSGA2 techniques, which are available in EASY, are presented below.

SPEA: SPEA was proposed by Zitzler and Thiele (188) in 1998 as a MOO cost assignment technique based on Pareto dominance. In SPEA, $\Phi(\vec{x})$ is computed in two steps as follows:

Step 1: (Strength Computation) The strength of each member of the population $P = P_\lambda^g \cup P_\mu^g \cup P_e^{g-1}$ is computed. The strength of individual i (S_i) is defined as the sum of the population members which are dominated by i , divided by the population size, namely

$$S_i = \frac{\sum(j : j \in P \wedge i \prec j)}{\sum P}, \quad \forall i \in P = P_\lambda^g \cup P_\mu^g \cup P_e^{g-1} \quad (2.10)$$

Step 2: (Cost Computation) The cost value Φ for each population member is computed as the sum of strengths of the population members which dominate it,

2. Evolutionary Algorithms (EAs)

$$\Phi_i = \sum_{j \in P \wedge i \prec j} S_j \quad (2.11)$$

SPEA2: SPEA2 (187) is an enhanced variant of SPEA which additionally incorporates density information. A nearest neighbour density estimation technique which allows a more precise guidance of the search process is incorporated, thus improving, compared to SPEA, the distribution of individuals along the Pareto front.

SPEA2 performs three steps to compute Φ . These steps are:

Step 1: (Strength Computation) As in SPEA.

Step 2: (Density Computation) For each population member, the density metric D_i is calculated as a function of its distance from its closest neighbour in the objective space d_i ,

$$D_i = \frac{1}{d_i + 2} \quad (2.12)$$

where

$$d_i = \min(\| \vec{F}_i - \vec{F}_k \|), \quad k \in P$$

Step 3: (Cost Computation) The cost values for all population members are calculated as the sum of the raw cost R_i and the density metric, D_i ,

$$\Phi_i = R_i + D_i \quad (2.13)$$

Here, R_i is the sum of the individual strengths of the population members which dominate it, namely

$$R_i = \sum_{j \in P \wedge i \prec j} S_j$$

NSGA2: NSGA2 (36) is an improved variant of NSGA (170) addressing the high computational complexity of non-dominated sorting, the lack of elitism and the need for specifying the sharing parameter of the initial algorithm. A modified

2.3 The Evolutionary Algorithm SYstem (EASY)

variant of NSGA2 is available in EASY. As proposed in (36), instead of computing Φ values, NSGA2 relies upon sorting the individuals according to Pareto dominance and density criteria. The modified variant of NSGA2 used in this PhD thesis includes the following steps:

Step 1: (Front Ranking) All members of the population are classified in fronts with decreasing Pareto dominance.

Step 1-a: (Initialization) Initialize the front counter $i = 0$ and the auxiliary set $S = P_\lambda^g \cup P_\mu^g \cup P_e^{g-1}$.

Step 1-b: (Find non-dominated members of S) Locate the non-dominated members of S and copy them to \mathcal{F}_i .

Step 1-c: (Update) Update $S = S - \mathcal{F}_i$ and $i = i + 1$. if $S \neq \emptyset$ go to Step 1-b; else, set the number of fronts N_F equal to $i - 1$.

Step 2: (Density Computation) For each member $j \in \mathcal{F}_i$, the density metric d_j , which is equal to the average side-length of the cuboid defined by its neighbouring individuals in \mathcal{F}_i , is computed. This procedure is repeated for all fronts i and all individuals are given a density metric value.

Step 3: (Cost Computation) Based on the density metric (calculated in Step 2) and front i (calculated at Step 1) which each individual belongs to, a scalar Φ is assigned to each individual as follows,

Step 3-a: (Initialization) Set $i = 0$ and $\Phi_b = 1.0$.

Step 3-b: (Computation of Φ) For each member $j \in \mathcal{F}_i$, Φ is calculated as

$$\Phi_j = \Phi_b \left(1 + \frac{d_{max} - d_j}{d_{max}}\right)$$

where d_{max} is the maximum of all $d_k \forall k \in \mathcal{F}_i$

Step 3-c: (Update) Update $\Phi_b = 1.1 \max(\Phi_j)$ and $i = i + 1$. If $i \leq N_F$ return to Step 3-b, otherwise stop.

2.3.3 Constraints' Handling in EASY

In EASY, penalty functions are used to handle constrained optimization problems (COP), see section 2.2. Penalty terms, proportional to the magnitude of the constraint violation, are added to the cost function. According to eq. 2.14, for each constraint, a “nominal threshold” d_k is defined as the maximum possible

2. Evolutionary Algorithms (EAs)

constraint function value allowed by the optimization problem. Furthermore, a “relaxed” threshold $d_k^* > d_k$ is introduced. Once an individual violates the relaxed threshold of one of the constraints ($c_k > d_k^*$), death penalty ($\Phi = \infty$) is given to it. If the individual violates one or more of the constraints ($d_k < c_k < d_k^*$) without exceeding any of the relaxed threshold values, its cost value Φ is penalized as follows:

$$\Phi(\vec{x}) = \Phi(\vec{x}) + \prod_{k=1}^K \begin{cases} \exp(a_k \frac{c_k(x) - d_k}{d_k^* - d_k}) & , c_k(x) > d_k \\ 1 & , c_k(x) \leq d_k \end{cases} \quad (2.14)$$

where the user-defined set of coefficient values in \vec{a} control the penalization intensity.

2.3.4 Metamodel-Assisted Evolutionary Algorithm (MAEA)

Due to the use of costly evaluation models (such as the CFD software to numerically predict flows in or around complex 3D geometries), solving engineering optimization problems by means of EAs may become very computationally demanding. The extensive smart use of low-cost surrogate evaluation models (often referred to as “metamodels”) during the optimization decreases substantially the number of calls to the computationally expensive, problem-specific evaluation code (CFD). This makes EAs a viable tool which can routinely be used to solve large-scale industrial optimization problems, in affordable wall clock time. Literature surveys on the use of metamodels within EAs can be found in papers (45, 47, 69, 90) or books (103).

Polynomial regression, artificial neural networks, Gaussian processes etc. have all been used as metamodels. The existing MAEAs differ since they apply different interactions between the metamodel and the problem-specific evaluation models. Hereafter, all of them will be referred to as MAEAs. In this thesis, on-line trained metamodels are used (69, 71, 99).

In on-line trained metamodels, for all but the first few generations, the metamodels are used to pre-evaluate the current population. Based on the outcome of approximate pre-evaluations (Inexact Pre-Evaluations, IPE), the few most promising individuals are identified and these solely undergo evaluation with the problem-specific evaluation model to compute their “exact” objective function value(s), before proceeding to the next generation. The (μ, λ) MAEA is sketched in fig. 2.10.

Radial Basis Function (RBF) Networks: In this PhD thesis, RBF networks are used as metamodels. Nevertheless, the new methods proposed in the next

2.3 The Evolutionary Algorithm SYstem (EASY)

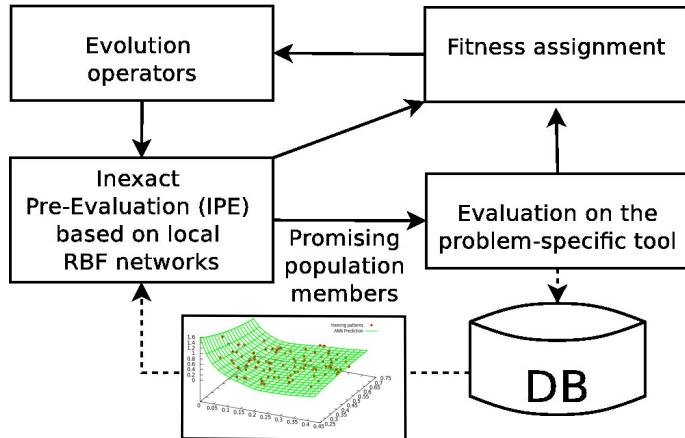


Figure 2.10: The (μ, λ) MAEA using on-line trained metamodels. The loop corresponds to a single EA generation. The IPE phase, within the EA-based search, is described in (69, 71, 99).

chapters could be combined with any other type of metamodels. RBF networks are artificial neural networks (ANNs) (82) with three neuron layers, (input, hidden and output) as shown in fig. 2.11. Signals propagate through the network in the forward direction, from the input to the output layer, by performing a nonlinear mapping (eq. 2.15) followed by a linear one. This mapping introduces weight coefficients w_l that must be computed during the network training on a number of available patterns. An RBF network to be used within a MAEA should have N input units, i.e. as many as the design variables. The hidden layer includes L nodes, associated with the so-called RBF centers c^l . At each hidden neuron, a nonlinear mapping of the input signals to a single value is performed using the radial-basis activation function $G : \Re^N \rightarrow \Re$, acting on the distance of input \vec{x} (eq. 2.15) from the corresponding center $\vec{c}^l \in \Re^N$. In the present thesis, the Gaussian activation,

$$G(u, r) = \exp\left(\frac{-u^2}{r^2}\right) \quad (2.15)$$

where $u = \|x - c^l\|_2$ the distance from the corresponding l^{th} RBF center, is used.

The radii or widths r values may considerably affect the prediction abilities of the network; these are computed using heuristics, (82). The output layer includes as many nodes as the expected network responses. The single response we are dealing with is expressed by the sum of the weighted output signals from the

2. Evolutionary Algorithms (EAs)

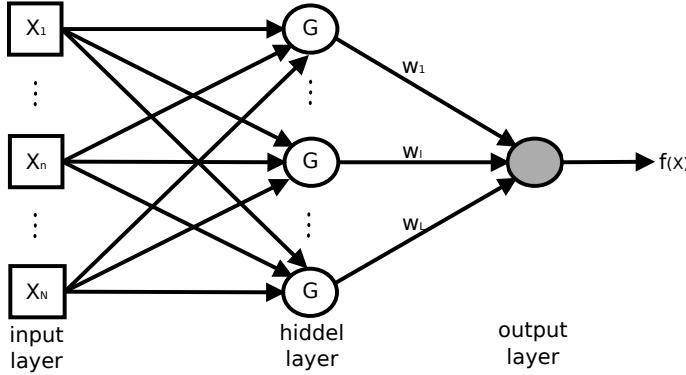


Figure 2.11: Radial basis function (RBF) network.

hidden neurons, as follows

$$f(x) = \sum_1^K w_i G(u(x_i), r) \quad (2.16)$$

If the number of hidden nodes (K) is equal to the number of training patterns (T), the choice of the *RBF* centers is straightforward, i.e. $\vec{c}^{(t)} \equiv \vec{x}^{(t)}$, $t \in [1, T]$, ensuring that the T samples are exactly interpolated. In this case, the network training requires the solution of the $T \times T$ symmetric linear system of equations,

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

A common way to increase the network's generalization (144, 175, 176) is by using a smaller number of appropriately selected hidden nodes than the training patterns, i.e. by selecting $K < T$. In such a case, the selection of the *RBF* centers is not as straightforward as in the aforementioned case and is of great importance because it strongly affects the prediction ability of the network. In (99), a selection scheme for the *RBF* centers based on self-organizing maps (*SOMs*), (59, 82) was proposed. This practically consists of a two level learning process, namely the unsupervised and supervised ones. During the unsupervised learning, through their standard processes: competition, cooperation and adaptation, *SOMs* classify the training patterns into K clusters. Each cluster gives a single *RBF* center $\vec{c}^{(k)}$, considered to be representative of the cluster as a whole. The corresponding radius r_k is computed using heuristics based on distances between the centers, (20, 82, 99, 102). During the supervised learning, the synaptic weights are computed by minimizing the approximation error of the *RBF* network over the training set, while considering smoothness requirements.

2.3 The Evolutionary Algorithm SYstem (EASY)

The so-called *Importance Factors (IFs)*, as proposed by Giotis et al. (71), can improve the prediction ability of the *RBF* network. Incorporating N extra coefficients ($I_n, n = 1, N$), the importance of each design parameter on the network response is quantified and exploited in order to increase the overall *RBF* network prediction quality. High I_n values indicate high sensitivity of the objective function in the vicinity of the design point under consideration with respect to the n -th input variable. The *IFs* are computed by the *RBF* network as a by-product of the training process and are used to improve the quality of networks' output. In detail, each time an improved solution is computed by the *EA* (index b , which stands for the current best), a local *RBF* network is built and N partial derivatives $\partial o^{(b)} / \partial x_n$ are computed using the closed-form response expressions; by doing so, $\partial o^{(b)} / \partial x_n$ become the exact derivatives of an approximate function. Based on these derivatives, a weighted norm is introduced and used instead of the standard one (i.e. instead of eq. 2.16), as follows

$$\|\vec{x}^{(t)} - \vec{c}^{(k)}\|_{wei} = \sqrt{\sum_{n=1}^N I_n \left(x_n^{(t)} - c_n^{(k)} \right)^2}, \text{ where } I_n = \frac{\left| \frac{\partial o^{(b)}}{\partial x_n} \right|}{\sum_{i=1}^N \left| \frac{\partial o^{(b)}}{\partial x_i} \right|} \quad (2.17)$$

2.3.4.1 Inexact Pre-Evaluation Algorithm

The algorithm starts as a conventional $(\mu, \lambda)EA$ (by exclusively making use of the exact evaluation model) for the few starting generations until a user-defined minimum number of previously evaluated individuals enter the *DB*. Then, the *IPE* phase starts and, in subsequent generations, instead of evaluating the offspring population on the costly problem-specific model, the following actions are taken:

Inexact Evaluation: For each offspring, $\vec{x} \in P_{\lambda,g}$, the objective function values $\tilde{\vec{F}}(\vec{x})$, are approximated using a local metamodel trained on a small number of data selected from the *DB*.

Screening: Based on the $\tilde{\vec{F}}(\vec{x})$ values, a provisional Φ value (denoted by $\tilde{\Phi}$) is assigned to each offspring.

Exact Evaluation: For all $\vec{x} \in P_e$, the “exact” objective function values $\vec{F}(\vec{x})$ are computed and stored in the *DB*. Practically, this step determines the *CPU* cost of each generation.

2. Evolutionary Algorithms (EAs)

2.3.5 Hierarchical Evolutionary Algorithms

As mentioned in section 2.1, an additional way to reduce the optimization turn-around time is by using the so-called hierarchical EA (HEA). HEAs are based on a hierarchical search structure, which consists of a number of semi-autonomously “evolving” levels (multi-level structure). Within each level, different evaluation tools, search techniques and/or problem parameterizations can be used, (39, 72, 83, 93, 94, 95, 97, 101, 166). Each level solves a different variant of the same problem. Furthermore, adjacent levels may exchange their best individuals according to one- or two-way inter-level communication schemes. The following three hierarchical schemes/modes, schematically presented in fig. 2.12, can also be combined in a single scheme.

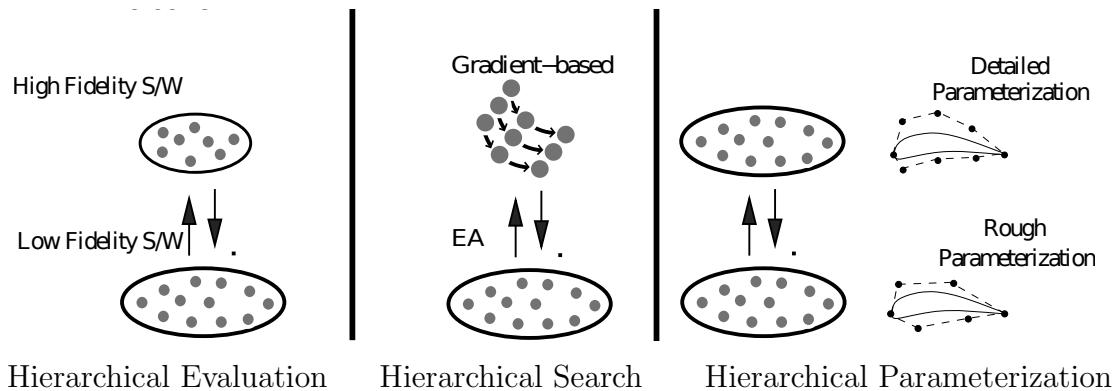


Figure 2.12: HEAs. Schematic representation of three hierarchical schemes.

Left: Hierarchical Evaluation. Middle: Hierarchical Search. Right: Hierarchical Parameterization. All this types can be combined or used separately, at will, utilizing two or more levels.

The three hierarchical schemes are described below:

(a) Hierarchical Evaluation Scheme: The hierarchical evaluation scheme allows the exploitation of a number of different evaluation tools/softwares with different fidelity and cost (computational and/or economical). Each tool is assigned to a different EA or MAEA level. By convention, the lower level undertakes the exploration of the design space by utilizing the low-cost and reduced fidelity tool so as to locate near-optimal solutions at low CPU cost and, then, delivering them to the higher level(s) for further refinement. On the higher level(s), being exploitation-oriented and responsible for delivering optimal solution(s), evaluation tools of higher fidelity and CPU cost are employed. Two-way inter-level communication can be used. So, apart from the migration of promising individuals upwards for them to undergo refinement, higher level individuals may also

2.3 The Evolutionary Algorithm SYstem (EASY)

move downwards so as to stimulate a more exhaustive search in their neighborhood, based on a low-cost evaluation tool. In the hierarchical evaluation scheme, as many DBs as the number of evaluation tools should be maintained. In a two-level algorithm, for instance, the high-fidelity DB, recording entries from evaluations based on the high-level problem-specific tool, is linked to the high level and the associated IPE phase. Next to it, a low-fidelity DB will do the same on the low level. Without loss in generality, differently configured MAEAs can be used on each level. An example of hierarchical evaluation used to handle an industrial application, is presented in fig. 2.13, where the higher level is associated with a Navier-Stokes equations' solver and the lower level with an Euler one.

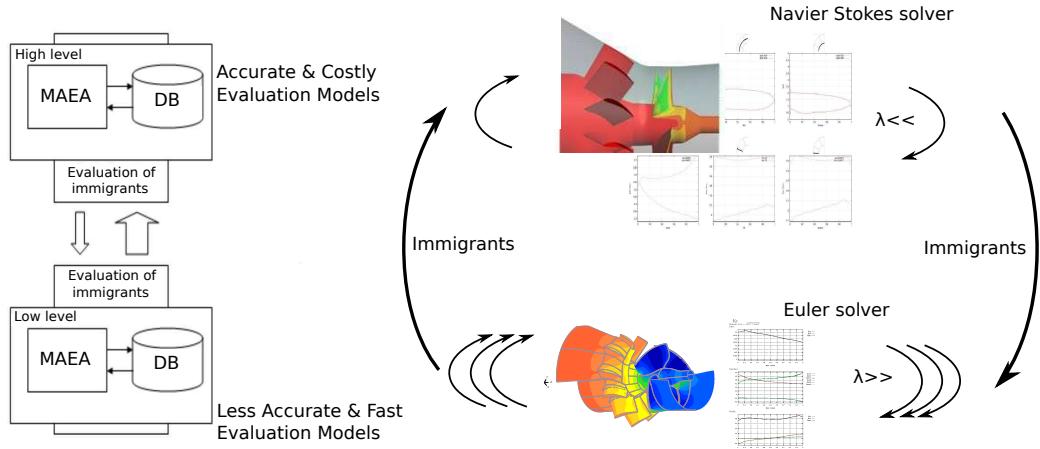


Figure 2.13: Left: Schematic representation of a two-level HMAEA (Hierarchical MAEA). The high level is associated with the expensive and accurate model and the lower level with the cheap and, thus, less-accurate one. Communication between the two level is possible in both directions including re-evaluation with the evaluation tool associated with receiving level. Right: Example of a HMAEA used in hydraulic turbine design (66).

(b) Hierarchical Search Scheme: In hierarchical search scheme, each level utilizes a different search technique and thus becomes able to combine the advantages of EAs with other search methods. In case the objective function gradient can be computed, the combination of EAs with gradient-based methods (GBM) is of great interest. Stochastic methods, such as EAs (DEAs, DMAEAs, etc.), are typically used for the exploration of the design space, on the lower level, exploiting

2. Evolutionary Algorithms (EAs)

their ability to escape local optima. Then, promising solutions migrate from the low level to its immediate higher one which uses a GBM to efficiently, with minimum turn-around time, locate the nearest optimum. The migration of promising solutions can be bi-directional, to accentuate the exploration capabilities of low-level EAs. The opposite configuration (GBM on the low and EA on the high level) is also possible. Hierarchical search requires particular attention if a GBM is employed to solve MOO problems, by seeking the front of non-dominated solutions. A common practice is to act on a fitness function formed by concatenating weighted objectives, which is appropriate only for convex fronts. Alternatively, as proposed in (93), the GBM can optimize the same scalar cost function (Φ) used by the EA, computed for instance via the SPEA2 technique, after appropriately approximating its non-differentiable functions. Should a GBM undergo the refinement on any level, methods for computing the gradient of the objective function must be available. In aerodynamic optimization, the computation of the gradient can be based on the adjoint method, (136), at extra cost which is approximately equal to that of solving the flow equations.

(c) Hierarchical Parameterization Scheme: In hierarchical parameterization different parameterizations are associated with each level. Typically a rough parameterization with a reduced number of design variables is associated with the lower level, where rough designs are quickly detected due to the small dimensionality of the problem. On the other hand, a detailed parameterization is associated with the higher level. The hierarchical or multi-level parameterization scheme should be configured in a way that makes all but the high level not affected (or, at least, slightly affected) by the curse of dimensionality. During the inter-level migration steps, a few promising solutions detected on the low level(s) are sent to the higher level(s) for refinement. This hierarchical scheme is suitable for shape optimization problems in which the design vector comprises the coordinates of control points of parametric curves and surfaces since, increasing or decreasing the number of control points via knot insertion and removal algorithms, allows exact (upwards) or approximate (downwards) transformations of the design vectors.

3

Knowledge-Based Design

It is common for all companies to keep archives of previous successful designs. As a consequence, irrespective of the design method adopted, there are often reasons for new designs to be based on “similar” previous ones, instead of starting the design procedure from scratch. As a matter of example, if for instance a new turbomachinery rotor is to be designed, a useful “similar” design might be a rotor successfully designed in the past for “slightly” different operating conditions and/or “slightly” different objectives or constraints. In some cases, these designs might have been performed enough time ago, using “old-fashioned” design procedures according to different parameterizations. In general, the term Knowledge-Based System (KBS) (6) is used to describe methods and processes for designing new products, engines, services etc. based on stored/archived knowledge which is available in the form of previous successful designs.

In this thesis, a new design method, to be referred to as Knowledge-Based Design (KBD), which combines EAs with principles and concepts from Knowledge-Based Systems (KBS) in order to create an automated design procedure, is proposed. It employs a KBS similar to Case-Based Reasoning (CBR) (113, 114, 153, 167), in order to utilise/exploit the availability of an archive of successful designs. In the design of aero- or hydrodynamic shapes, a pattern of similar geometries often occurs when dealing with similar problems. For example, the optimal airfoil, regarding objective(s) “A” and constraint(s) “B”, of a compressor cascade operating at conditions “C” and inlet flow angle α_1 (where “C” includes everything but α_1) will be, more or less, similar to the airfoil of a different compressor designed for the same “A”, “B” and “C” and inlet flow angle equal to $\alpha_1 + 1^\circ$. Therefore, if a design is optimal for a set of different/neighbouring conditions, it still holds valuable information about the design space that can be exploited in order to speed-up the design process. This is the main purpose of the proposed KBD method which shares the advantages of both EAs and KBS.

In more detail, from the CBR point of view, the proposed methodology

3. Knowledge-Based Design

achieves in creating a fully automated and time-efficient *revise* step, this is the step that adapts the existing “similar” designs so as to make them perform optimally at the new operating conditions and/or according to the new objectives. In this thesis, this is achieved by using an EA as a revise tool. In order to have a cost-efficient procedure, capable for being routinely used in an industrial environment, additional EA speed-up techniques, mainly the use of metamodels (MAEA, section 2.3.4), are employed. The straightforward way to achieve this goal would be to simply inject the “similar” archived designs into the initial population of an EA, by replacing some of the randomly generated population members. In contrast, the method proposed in this thesis utilizes statistical analysis to introduce a new set of design variables, based on the archived designs, in order to further speed-up the EA itself. This is achieved through:

- a) The significant reduction in problem dimension (i.e. the number of design variables handled by the EA), thus improving both the EA (faster convergence to the optimal solution) and MAEA efficiencies (use of ANNs as metamodels with greater prediction accuracy and without delaying the start of the IPE phase, as described in section 2.3.4).
- b) The automatic definition of the newly introduced design variables’ (to be referred to as optimization variables, to distinguish them from the design variables used to parameterize the geometry) range; the designer overcomes the burden of arbitrarily guessing their lower and upper bounds and, therefore, eliminates the possibility of accidentally excluding search sub-areas corresponding to the optimal design.
- c) The association of degrees of importance to the optimization space regions. Practically, higher probability to host candidate solutions is given to the regions of the design space pinpointed by the statistical analysis of the archived “similar” designs, based on a pre-defined probability distribution. This allows the exploration of extended design spaces, with different levels of importance associated with different sub-regions, according to statistics.

3.1 Knowledge-Based Systems (KBS)

KBS (6) are systems based on methods and techniques of artificial intelligence. Their core components are the knowledge base and inference mechanisms. In this thesis, a KBS named Case-Based Reasoning (CBR) (113) is combined with EAs, in an optimal way, giving rise to the KBD method. For reasons of clarity, a brief introduction to the CBR method follows, before presenting the proposed KBD method.

3.1.1 Case-Based Reasoning Method - Principles

CBR, (113, 114, 153, 167), can be seen as a problem solving method that reuses past cases and experience to conceive a solution to the current problem, whereas other major artificial intelligence techniques rely on mapping generalized relationships between problem descriptions and conclusions. CBR has the advantage of utilizing experience based on past successful problem solutions. The main tasks of a CBR system is to identify the problem in hand, find one or more similar past case(s), use this information to suggest a solution to the current problem and update the system by learning from this experience.

The origins of CBR dates back to the work by Schank and Abelson in 1977, (162). They proposed that general human knowledge about situations is recorded as scripts that allow extracting expectations and inferences. Scripts were proposed as structures for conceptual memory describing information about stereotypical events. However, experiments showed that this is not a complete theory of memory representation since people often confuse events with similar scripts. These conform with concept formulation, problem solving and experimental learning theories within philosophy and psychology, (168, 177).

Schank continued to investigate the role of previous situations (i.e. cases) and situation patterns in both problem solving and learning, (161). Simultaneously, Gentner, (61), developed a theoretical framework for analogy also relevant to CBR. Significant references to CBR can also be found in Wittgensteins observation, (183), that natural concepts are in fact polymorphic and cannot be classified by a single set of necessary and sufficient features; instead they can be defined by a set of instances (i.e. cases) with family resemblances. The latter work has been cited as the philosophical basis for CBR by Aamondt and Plaza, (5).

Though the roots of CBR could be claimed by several scientists, it was Schank and his co-workers who, in the early 80's, developed a cognitive model which the first CBR applications were based upon. Kolonder developed the first CBR system named CYRUS, (110, 111), which was an implementation of Schank's dynamic memory model. Later, its case-memory model served as the basis for several other CBR systems and was used in various disciplines ranging from law, (112, 154), to civil engineering, (130, 181).

The classical definition of CBR was coined by Riesbeck and Schank, (153): "*A case-based reasoner solves problems by using or adapting solutions to old problems*". This clearly defines what a CBR system can do, without however determining the way of doing it. This can be graphically understood by the so-called CBR-cycle (fig. 3.1). The CBR-cycle is based on four logical steps, as defined by Aamondt and Plaza, (5), which are often referred to as the four REs:

- RETRIEVE the most “similar” case(s),

3. Knowledge-Based Design

- REUSE the retrieved case(s) for the problem in hand,
- REVISE the case(s), if necessary, and
- RETAIN the new solution in the archive for future use.

The four REs can be represented by a schematic cycle, fig. 3.1.

In the proposed KBD method, the REVISE step is undertaken by an EA which automatically combines the RETRIEVED cases without requiring human intervention. Recall that the traditional CBR method requires a human expert to undertake task.

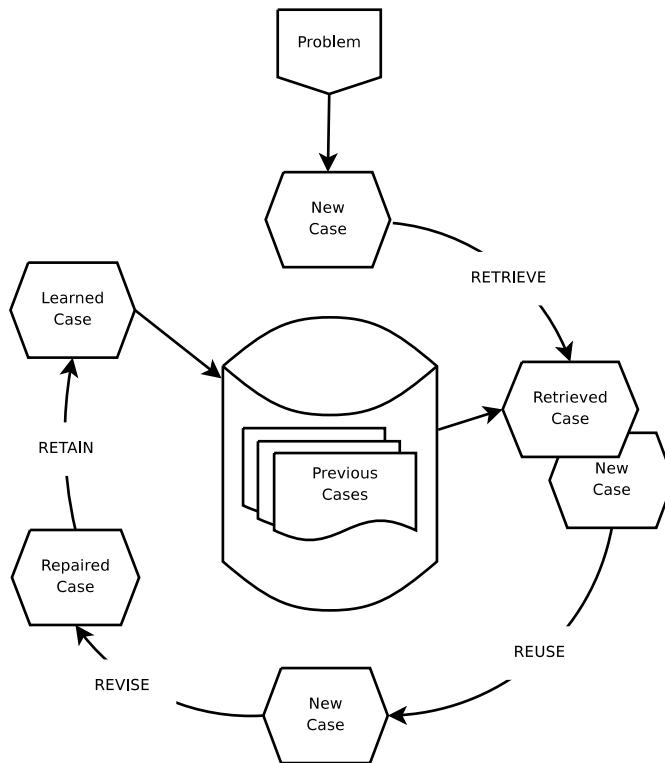


Figure 3.1: Schematic representation of the CBR-cycle; from (5).

3.2 The KBD Method

In order to have a fully automated REVISE step, a set of adaptation rules must replace human intervention. These rules take into account the deviations of the proposed solution from the desired one and suggest an improved proposal. The

rules governing the evolution of species ideally fit the above description. So, in this thesis, an EA is proposed to undertake the REVISE step.

Therefore, the present method aims at extending the optimization method by making it capable to accommodate and exploit pieces of useful information archived during previous relevant successful designs. The KBD method is much more elaborated than the simple and straightforward injection of retrieved designs in the initial population of an EA, where they replace some of the randomly generated ones. Instead, the KBD method introduces a new set of unknowns, the so-called optimization variables, to be used by the EA, instead of the design variables. In case of complex 3D geometries, such as those involved in the field of thermal and hydraulic turbomachines, the number of design variables is much greater and this slows the design-optimization process down. The optimization variables are introduced by expressing the candidate solutions as points in the optimization space, which is the space defined by the design vectors of the retrieved geometries as vector bases. The role of the optimization algorithm is to find the value set (or sets, for more than one objectives, in problems where the Pareto front of non-dominated solutions is sought) of the optimization variables which yields optimal performance. Since the number of optimization variables is much less than the number of design variables, the number of degrees of freedom is reduced and the EA or MAEA CPU cost is expected to be much lower.

3.2.1 Optimization Variables's Definition

In this section, the optimization variables introduced by the proposed KBD method are presented. Assume that a small number (m) of previous designs are retrieved from the archive. These designs must correspond to similar problems, as already discussed. For the proposed method, it is necessary for the retrieved geometries to be expressed in terms of the same parameterization. If techniques able to transform a parameterization scheme to any other (within a degree of accuracy, of course) are available, this task is straightforward. Let us denote by $GEO_i = (x_1^i, x_2^i, \dots, x_N^i)$, $i=1, m$ the m archived designs, with N design variables each.

The set of optimization variables can be defined, without yet associating any degree of importance to the optimization space subregions, by assuming that each new design x_j^{new} results from the combination of the m archived designs, weighted by the w_i ($i = 1, m$), or

$$x_j^{new} = \frac{\sum_{i=1}^m w_i x_j^i}{\sum_{i=1}^m w_i}, \quad j = 1, \dots, N \quad (3.1)$$

3. Knowledge-Based Design

Without loss in generality, one may assume that $w_i \in [0, 1]$.

Next step is the association of degrees of importance to the optimization space regions. This is achieved by rewriting eq. 3.1 as

$$x_j^{new} = \Phi_j^{-1}\left(\frac{\sum_{i=1}^m w_i \Phi_j(x_j^i)}{\sum_{i=1}^m w_i}\right) \quad (3.2)$$

through the introduction of the sigmoid cumulative distribution function Φ , (105), according to a user-defined probability distribution function. If the normal distribution function is used, Φ is given by

$$\Phi_{\mu\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x \exp\left(-\frac{(u-\mu)^2}{2\sigma^2}\right) \quad (3.3)$$

where μ is the mean value and σ the standard deviation of each design variable (j). Schematically,

$$\begin{pmatrix} x_1^1 \\ \vdots \\ x_N^1 \end{pmatrix} \begin{pmatrix} x_1^i \\ \vdots \\ x_N^i \end{pmatrix} \begin{pmatrix} x_1^m \\ \vdots \\ x_N^m \end{pmatrix} \rightarrow \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_N \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_N \end{pmatrix} \quad (3.4)$$

Since the m retrieved designs are usually located “around” the desired one in the operating conditions space and based on the assumption that this information is, more or less, transferable to the design space, the normal probability distribution is used to assign degrees of importance throughout this thesis. In a simplified example, the above statement says that: Assume that airfoil A with design variables vector \vec{x}_A is optimal, regarding objective(s) “O”, when operating at conditions “C”. Assume, also, that at these conditions the flow turning is equal to $\Delta a = 40^\circ$. Also, that design B, or \vec{x}_B , is optimal regarding the same objectives at the same conditions and delivers flow turning equal to $\Delta a = 41^\circ$. Considering the same objectives “O” and the same conditions “C”, if the optimal airfoil which delivers flow turning equal to $\Delta a = 40.5^\circ$ is sought, the optimal design N, or \vec{x}_N , would, most probably, be close to $\frac{\vec{x}_A + \vec{x}_B}{2}$ which is the vector of the mean values of the design variables of the retrieved geometries A and B. It is reminded that, if the normal probability distribution is used for importance assignment, the regions near the mean value of each design variables will be assigned the highest importance and, therefore, highest probability to accommodate candidate solutions. In case the user knows in advance that, for this particular problem, the above statement is not true, he might exchange the normal distribution with the appropriate probability distribution of his choice.

3.2 The KBD Method

The use of the cumulative distribution function of the normal probability distribution, eq. 3.3, practically assumes the bounds of b_i to be within $\mu_i \pm 3\sigma_i$, (105). To overcome this limitation, a single extrapolation variable Ψ which multiplies all σ values computed based on the archived designs and extends the search space outside $\mu_i \pm 3\sigma_i$, is optionally introduced. Ψ is multiplied with the computed standard deviations to yield the ones used in eq. 3.3, as follows

$$\begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_n \end{pmatrix} = \Psi \begin{pmatrix} \sigma_1^{computed} \\ \vdots \\ \sigma_n^{computed} \end{pmatrix} \quad (3.5)$$

If the available archived designs are not that many, which often is the case, then m is a significantly small integer. With either eq. 3.1 or 3.2, an optimization problem with only m weights as unknowns may lose its flexibility. Such a method may overcome the curse of dimensionality (since the number of optimization variables is noticeably lower than N) but may lead to sub-optimal solutions. To overcome this, the grouping of those design variables which are “co-related”. Relevant design variables such as, for instance, those defining the mean camber surface angle at the leading edge etc. are grouped together. After forming these groups, a different weight is associated with each one of them. The new weights are denoted by $w_{i,k}$, where the first index corresponds to the i^{th} design basis and the second one to the k^{th} group of design variables (where b_i belongs to). Finally, instead of either eq. 3.1 or 3.2, the expression

$$x_j^{new} = \Phi_j^{-1}\left(\frac{\sum_{i=1}^m w_{i,k} \Phi_j(x_j^i)}{\sum_{i=1}^m w_{i,k}}\right), \quad k = f(j) \quad (3.6)$$

can be used.

Based on eq. 3.6, an optimization problem with mK unknowns (or $mK+1$, to also account for Ψ), where K is the number of design variable groups, is formulated. Population-based search methods, such as EAs (or MAEAs), with the proposed parameterization, may locate the global optimum much more efficiently than an EA (or MAEA) based on the conventional parameterization. This is demonstrated on the solution of thermal (section 3.3) and hydraulic (section 5.3) turbomachinery design-optimization problems.

3. Knowledge-Based Design

3.3 Design of a Compressor Cascade using KBD

The design of a 2D compressor cascade operating at $M_1 = 0.54$, $a_1 = 44^\circ$ and $Re = 4 \times 10^5$ (Reynolds number based on the chord) for minimum total pressure losses is sought. Losses are measured in terms of the total pressure loss coefficient

$$\omega = \frac{p_{t1} - p_{t2}}{p_{t1} - p_1} \quad (3.7)$$

where indices 1 and 2 denote the cascade inlet and outlet, respectively. The blade airfoil is designed subject to a number of aerodynamic and geometrical constraints: the optimal airfoil must turn the flow by more than 30° and its thickness at three chordwise positions ($0.3c$, $0.6c$ and $0.9c$) must be greater than $0.10c$, $0.08c$ and $0.01c$, respectively.

The airfoil shape is parameterized based on the mean-camber line shape and thickness distributions (separately for the pressure and suction sides). All of them are parameterized using NURBS. The mean-camber line parameterization “introduces” three degrees of freedom, namely the leading edge (LE) angle, the trailing edge (TE) angle and the weight associated with the second control point. The suction side thickness distribution is controlled by 9 design variables, namely the x and y coordinates as well as the weights of three, out of the five, internal NURBS control points. The first and last control points are fixed at the predefined LE and TE positions. The pressure side thickness distribution is controlled via 15 design variables, standing for the x and y coordinates as well as the weights of five, out of the eight, internal control points. The first and last control points are fixed at the predefined LE and TE positions and the location of the second point is determined by the requirement of first-order continuity at the LE. The total number of design variables describing each candidate solution is 27, fig. 3.2.

The KBD method will be assessed through the comparison of a traditional MAEA with a KBD-MAEA. For both of them, $\mu = 20$, $\lambda = 60$ and $\lambda_e = 6$. Both cases use local RBF networks as metamodels, which are trained on a small number of training patterns (min. 10; max. 20 patterns). For the KBD-MAEA, the IPE phase starts after the first 100 individuals are stored in the DB of previously evaluated individuals. On the other hand, for the MAEA run, 400 individuals must be stored in the DB before the IPE phase starts. The earlier start of the IPE phase during the KBD-MAEA, as opposed to the MAEA, is due to the reduced number and range of the optimization variables. The KBD-MAEA uses 13 instead of 27 design variables which would be the case if the standard shape parameterization was used. The evaluation software is an integral boundary layer method, (42). The maximum number of calls to the evaluation software is 1500 in all cases, so as to compare them based on the same CPU cost; this is the only

3.3 Design of a Compressor Cascade using KBD

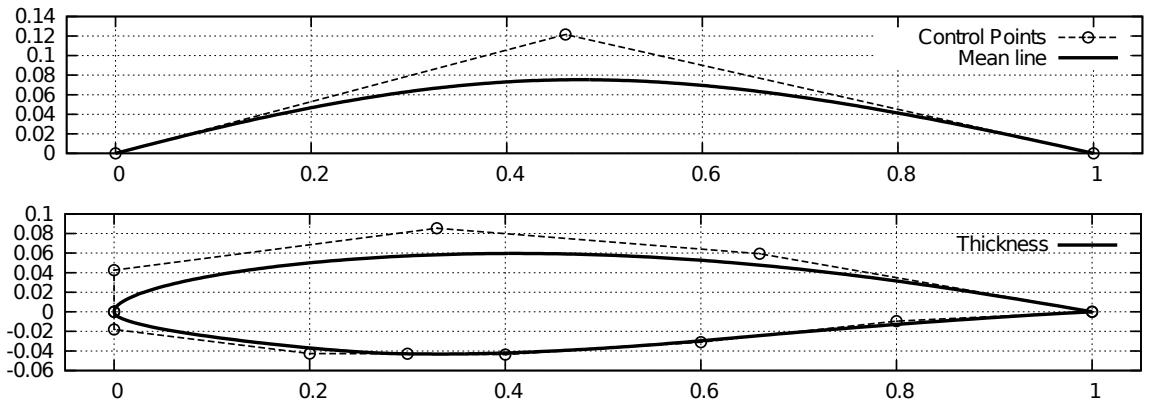


Figure 3.2: Design of a compressor cascade using KBD: Top: parameterization of the mean-camber line with 3 control points. Bottom: parameterization of the thickness distributions with NURBS using 5 control points for the suction side and 8 for the pressure side. The total number of design variables is 27.

termination criterion used.

Four archived designs were retrieved and used as design space basis vectors in order to define the optimization variables used by the KBD method. The archived designs have been designed for operation at different but neighbouring conditions, namely:

- **Archived design B_1 :** designed for $M_1 = 0.55$ and $a_1 = 45^\circ$, fig. 3.3.
- **Archived design B_2 :** designed for $M_1 = 0.50$ and $a_1 = 40^\circ$, fig. 3.4.
- **Archived design B_3 :** designed for $M_1 = 0.52$ and $a_1 = 42.5^\circ$, fig. 3.5.
- **Archived design B_4 :** designed for $M_1 = 0.58$ and $a_1 = 48^\circ$, fig. 3.6.

At their design conditions (see above) the archived geometries perform as follows:

- **Archived design B_1 :** $\omega = 0.01976$ and $\Delta a = 37.1^\circ$.
- **Archived design B_2 :** $\omega = 0.01845$ and $\Delta a = 24.4^\circ$.
- **Archived design B_3 :** $\omega = 0.02027$ and $\Delta a = 36.3^\circ$.
- **Archived design B_4 :** $\omega = 0.01877$ and $\Delta a = 36.6^\circ$.

The four retrieved airfoils, operating at the new flow conditions ($M_1 = 0.54$, $a_1 = 44^\circ$), without any modification in their shape, perform as follows:

- **Archived design B_1 :** $\omega = 0.0207$ and $\Delta a = 36.3^\circ$.

3. Knowledge-Based Design

- **Archived design B_2 :** $\omega = 0.0278$ and $\Delta a = 27.8^\circ$.
- **Archived design B_3 :** $\omega = 0.0201$ and $\Delta a = 37.5^\circ$.
- **Archived design B_4 :** $\omega = 0.0237$ and $\Delta a = 33.1^\circ$.

From the aforementioned ω and Δa values, the necessity for a *REVISE* step becomes obvious since all four retrieved airfoils suffer from relatively high losses and some of them fail to respect the flow turning constraint at the new flow conditions. These four designs were injected in the starting population of the MAEA, either with or without KBD.

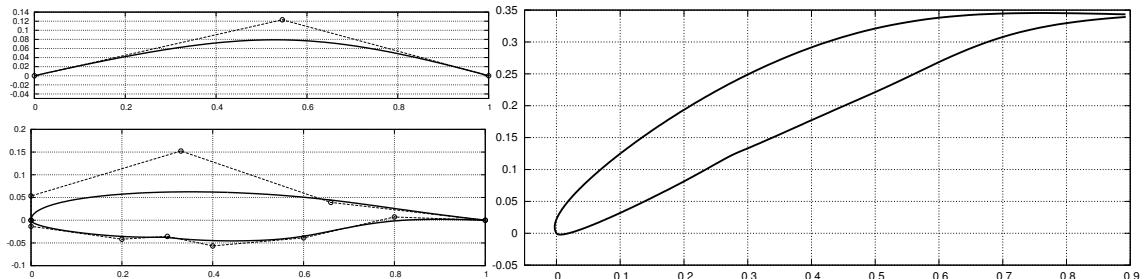


Figure 3.3: Design of a compressor cascade using KBD: Archived design B_1 .

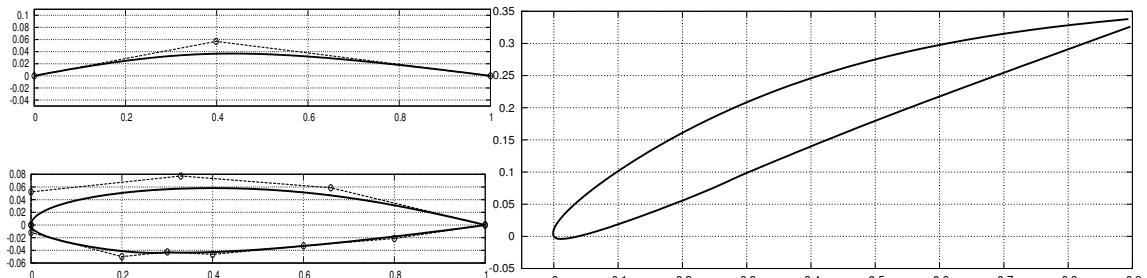


Figure 3.4: Design of a compressor cascade using KBD: Archived design B_2 .

Four different optimization runs were performed in order to assess the effect of the KBD method both on EAs and the implementation of metamodels in MAEAs. The corresponding convergence plots are shown in fig. 3.9. The use of metamodels in runs which do not rely upon the KBD method are not as efficient as expected. This is caused by the use of an extended design space which is really necessary for being able to reproduce all four archived geometries. To demonstrate this, the four archived geometries are plotted together in figs. 3.7 and 3.8. The large variance of the control points is more severe in the parameterization of the mean camber line (fig. 3.7) regarding both coordinates and, also, in the

3.3 Design of a Compressor Cascade using KBD

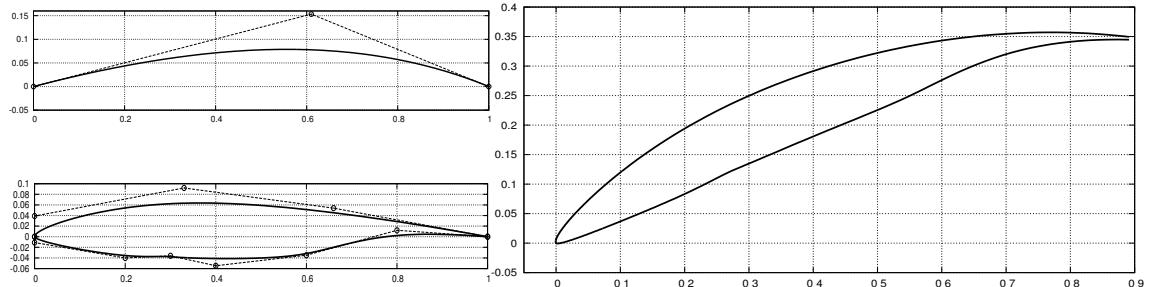


Figure 3.5: Design of a compressor cascade using KBD: Archived design B_3 .

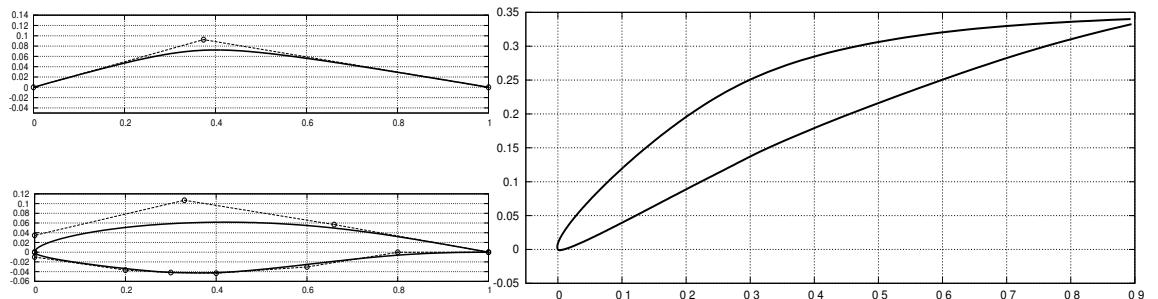


Figure 3.6: Design of a compressor cascade using KBD: Archived design B_4 .

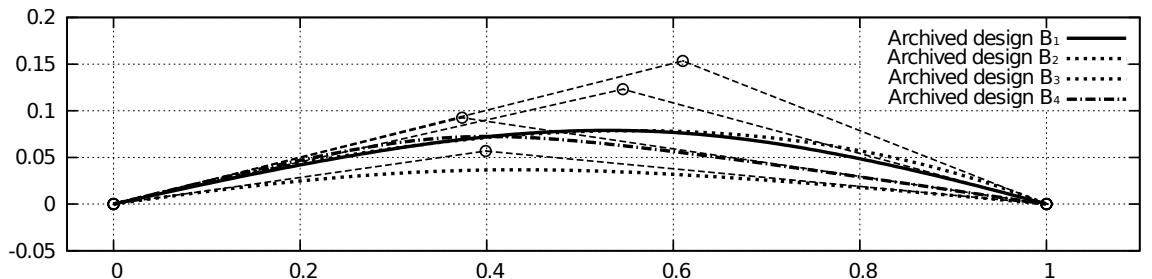


Figure 3.7: Design of a compressor cascade using KBD: Mean-camber lines of the four bases. The observed great variance recommends the use of very extended design variable ranges during the EA-based optimization. This significantly deteriorates EA efficiency and makes the use of metamodels questionable.

thickness distribution along the suction side (fig. 3.8). Using a more restricted design space would improve both the EA efficiency and the gain expected from the use of metamodels (MAEA) but could give suboptimal solutions if, by chance, the optimal geometry lays outside the design space. The comparison between EA and KBD-EA reveals the advantages of the proposed method (fig. 3.9), since faster convergence is achieved. The comparison of KBD-MAEA with KBD-EA shows that the KBD method enhances the performance of the metamodel-based pre-evaluation phase.

3. Knowledge-Based Design

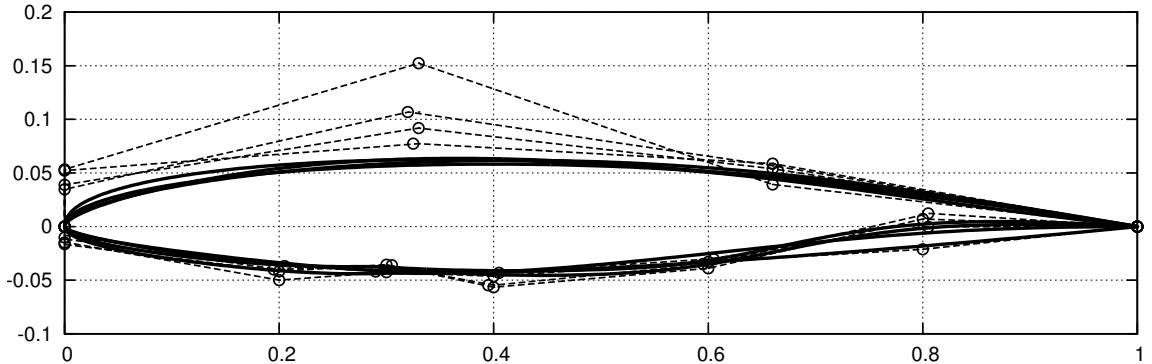


Figure 3.8: Design of a compressor cascade using KBD: Suction and pressure side thickness distributions and control point polygons for the four bases.

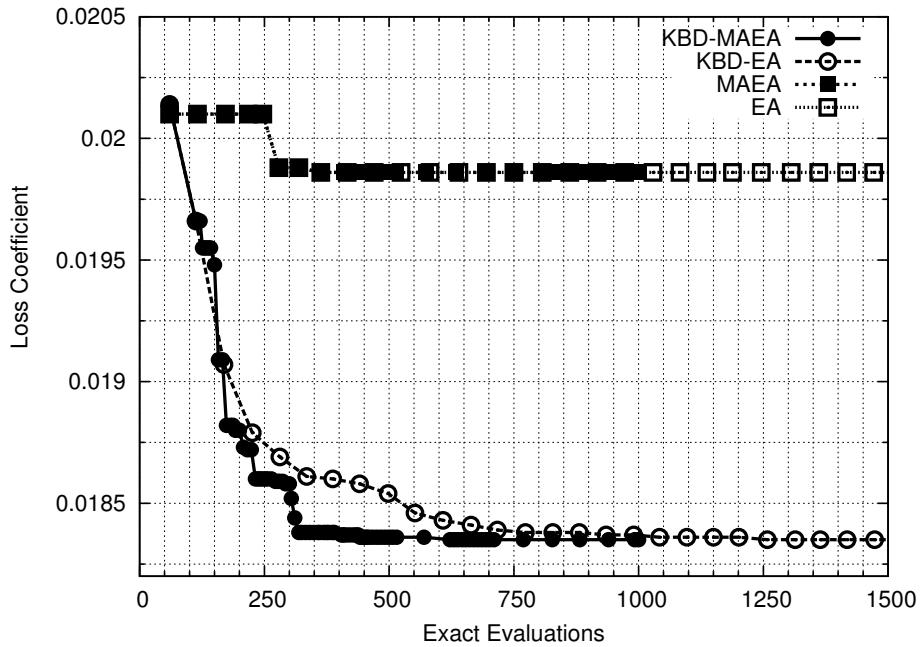


Figure 3.9: Design of a compressor cascade using KBD: Convergence comparison of four optimization runs using EA, KBD-EA, MAEA and KBD-MAEA. EA and MAEA have similar convergence plots, since the use of metamodels doesn't lead to extra gain, due to the extended range of the design variables. KBD-EA is significantly faster than both EA and MAEA, thus proving the advantages of the proposed method. KBD-MAEA performs even better thanks to the positive effect of the proposed method on the use of metamodels.

The optimal blade resulting from the KBD-MAEA is presented in fig. 3.10. This design delivers $\omega = 0.01834$ and $\Delta a = 30.2^\circ$ at the desired flow conditions, outperforms any of the design-bases for the same flow-conditions and respects

3.3 Design of a Compressor Cascade using KBD

all the constraints. The airfoil designed using the KBD-MAEA is of significantly better quality compared with those delivered by the conventional runs, with or without the use of metamodels.

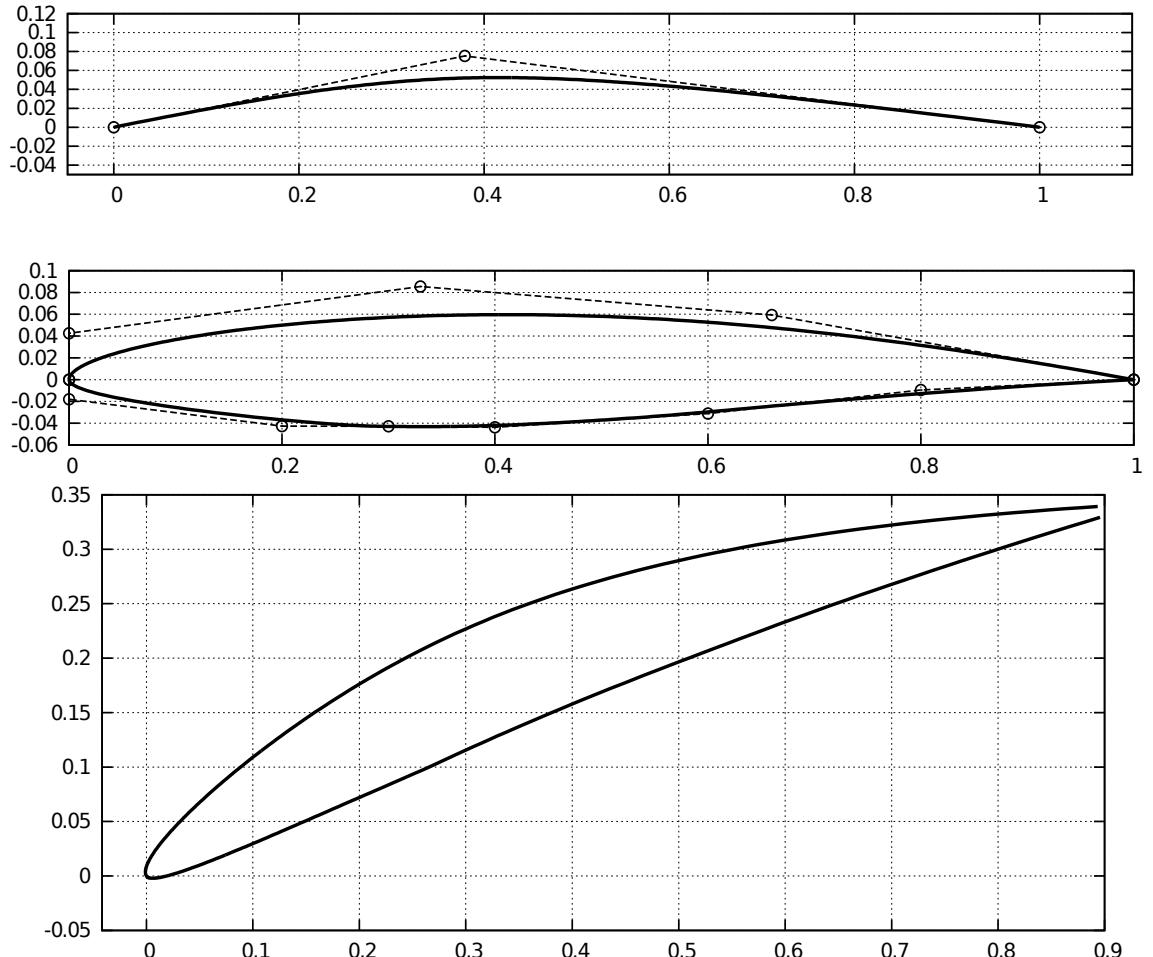


Figure 3.10: Design of a compressor cascade using KBD: The optimal airfoil resulting from KBD-MAEA. Top: mean-camber line and thickness distributions together with their control polygons. Bottom: The airfoil shape after superimposing the thickness distribution on the mean-camber line and rotated at the desired stagger angle.

3. Knowledge-Based Design

4

EAs and MAEAs Driven by Principal Component Analysis

This chapter investigates the reasons for the drop in EAs or MAEAs efficiency used to solve ill-posed optimization problems (67, 156, 158), especially when these possess a great number N of design variables, and proposes techniques to overcome this unpleasant situation. Any degradation in the EAs or MAEAs efficiency reflects upon the number of evaluations (on the CFD software, for instance) required to reach the optimal solution(s). Increasing the number of evaluations leads to longer optimization turn-around times and the routine application of EA-based optimization in large-scale industrial applications is, thus, hindered. The proposed methods adapt the application of the evolution operators accordingly and/or lead to the training of more dependable metamodels on patterns of properly reduced dimension (in case of MAEAs).

Modern industrial-scale optimization problems typically have several objectives and/or constraints with complex parameterization and a great number of design variables. The former is determined by the application under consideration whereas the latter is, often, “inherited” from the way experienced designers used to accomplish the same task in the past, without resorting to “sophisticated” optimization methods. From the practical point of view, a smooth transition to the EA-based optimization is necessary. Problems with several objectives and/or constraints which handle many design variables can easily become ill-posed, as discussed in section 4.1. The use of stochastic population-based search methods to solve ill-posed problems leads to excessively long computations, even on modern multi-processor platforms. Since the objectives and constraints are strictly defined by the application, the only way to increase the efficiency of EAs used to solve ill-posed optimization problems is via the appropriate handling of the parameterization.

Changing the parameterization, though, may have a number of negative ef-

4. EAs and MAEAs Driven by Principal Component Analysis

fects. First among them is that, in any industrial environment, a valuable amount of experience is embodied into the existing parameterization, in the sense that an experienced designer practically knows qualitatively what to change in order to achieve his/her goal; this would be lost by changing or modifying the parameterization scheme. Also, there is no guarantee that the new/altered parameterization, which must be built in accordance with the objectives so as to give a better-posed optimization problem, includes variables with clear interpretation (being, for instance, the coordinates of Bézier control points, etc.). In fact, this is highly unlikeable. Any new parameterization, with unclear interpretation of the design variables, makes the accumulation of new experience impossible. Furthermore, in a specific project, the best parameterization depends on its correlation with the objectives and constraints. Therefore, a different parameterization would be necessary for any new design-optimization procedure, based on its objectives. To make things even worse, the best parameterization cannot be known in advance, since one should have the Pareto front available (in MOO problems) so as to extract the most appropriate parameterization (see below, in this chapter). All the above can be overcome by implementing the techniques proposed in this chapter.

As stated in (158), standard GAs can find the global optimum in a number of widely used test functions due, not only to the efficiency of the recombination operator and the regular distribution of local minima, but also to the fact that the test functions are decomposable. Though (158) was based on GAs, its outcomes can be generalized to EAs. In (156, 157), it was stated that “real-life” engineering design optimisation problems, as opposed to the widely used mathematical test cases, also involve “interactions among design variables”. There, the notions of “inseparable function interaction” and “variable dependency” were introduced. The existence of “variable dependency” poses a number of challenges in MOO algorithms which are similar to what is referred to as ill-posed optimization problems. Also, as stated in (157), depending upon the nature of “variable dependency”, additional features such as bias (non-linearity), multi-modality, deception and discontinuity may also occur and these will cause additional loss in the optimization algorithm efficiency.

4.1 Ill-Posed Optimization Problems

The two main reasons for increasing the number of calls to the evaluation software required by an EA or MAEA, in order to locate the optimal solution(s), are: (a) the high number of design variables, $N >>$ and (b) the fact that an ill-posed optimization problem is solved. Needless to say that a high-dimensional ill-posed optimization problem is the worst case for an EA or MAEA. Unfortunately, in practice, most engineering optimization problems share both features mentioned

above .

In EAs, the high number of design variables (N) results in efficiency loss. In the best case (well-posed problem), the cost for computing the optimal design(s) increases linearly with N . In MAEAs, the presence of $N \gg$ design variables, apart from the way the optimal solution is approached (direct effect), impacts also the use of metamodels for which the training time and, mainly, their prediction error increase noticeably. The loss in efficiency due to the “less-accurate” metamodel predictions (indirect effect) is, practically, superimposed to the performance degradation caused by the direct effect. As a remedy to optimization problems with $N \gg$, the KBD method was proposed in Chapter 3. From a completely different point of view, the KBD method reduces the problem dimension by exploiting the knowledge embodied into a small number of available archived designs and has already been presented and validated.

On the other hand, ill-posed optimization problems are those dealing with (a) anisotropic and (b) non-separable objective function/s $f(\vec{x})$. Note that (a) is a prerequisite for (b). In fact, ill-posed optimization problems are those dealing with non-separable objective functions. The definitions of anisotropic and non-separable functions follow in sections 4.1.1 and 4.1.2, respectively. Ill-posed problems, in combination with $N \gg$, give rise to the so-called “curse of dimensionality”. This stands for the situation in which the cost for solving an optimization problem increases superlinearly, instead of just linearly, with N .

4.1.1 Anisotropic Objective Function

An objective function $f(\vec{x})$, $\vec{x} = (x_1, \dots, x_N)$ is anisotropic with respect to \vec{x} if it is differently affected by the same changes in the different components of \vec{x} . Though this thesis is dealing with non-gradient-based optimization methods, the anisotropy of f can be quantified by the variation of $\frac{\partial f}{\partial x_i}$ among the design variables i , close to the optimal solution.

Let us assume that the convex-quadratic objective function

$$f(\vec{x}) = \frac{1}{2} \vec{x}^T H \vec{x} \quad (4.1)$$

where H is a symmetric positive definite $N \times N$ matrix. The variations in $\frac{\partial f}{\partial x_i}$ are expressed by the condition number a of matrix H . In such a case, a is defined as the ratio of H ’s largest to smallest eigenvalue. These two eigenvalues correspond to the squared relative lengths of the principal axes of the ellipsoid $\vec{x}^T H \vec{x} = 1$. For points located along the principal axes, the gradient aligns with the axis itself and the ellipsoid-axis length determines the magnitude displacement in the design space required for obtaining a certain change in f . Therefore, f is

4. EAs and MAEAs Driven by Principal Component Analysis

said to be isotropic if the condition number of H is $a = 1$ (corresponding to a sphere or hyper-sphere-like isolines, see fig. 4.1, left) and anisotropic if $a \gg 1$ (corresponding to an ellipse-like isolines, see fig. 4.1, right).

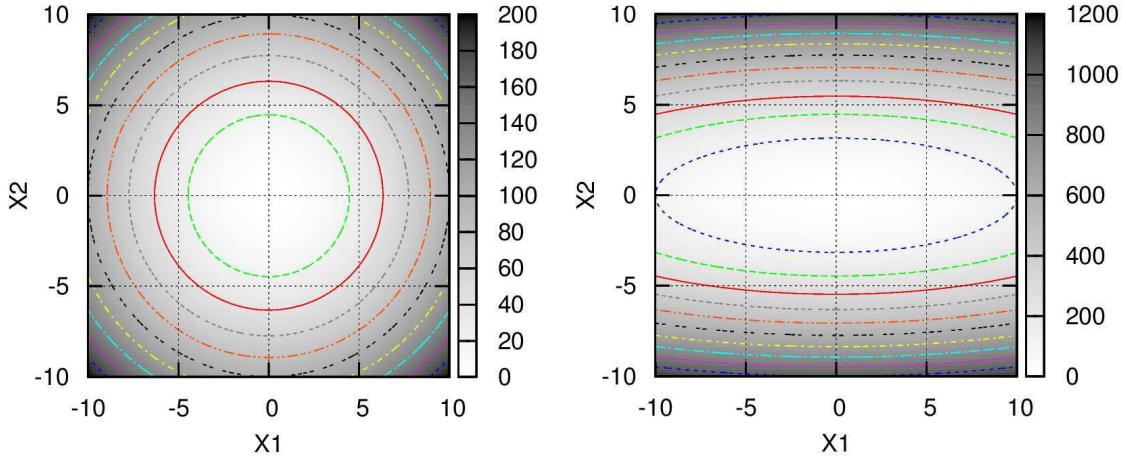


Figure 4.1: Iso-value contours of an isotropic objective function with two design variables (left); the impact of x_1 and x_2 on f is the same. Iso-value contours of an anisotropic objective function, in which x_2 has greater impact on f than x_1 (right).

In practice, EAs are capable to efficiently deal with anisotropic objective functions without damaging their efficiency. If, however, the objective function is anisotropic and non-separable (see definition in section 4.1.2), the EA or MAEA is expected to perform less efficiently, unless a particular treatment is employed; such a treatment is proposed in this chapter.

Furthermore, the anisotropy of f leads to the notion of more and less important design variables (or, in the case of non-separable f , more and less important directions in the design space). This piece of information can be used to reduce N , via the truncation of less important variables during the metamodel training phase, so as to increase its prediction ability.

4.1.2 Non-Separable Objective Function

The separability of an objective function $f : \vec{x} \mapsto f(\vec{x})$ is defined separately for each and every variable $x_i \in \vec{x}$. An objective function is said to be separable with respect to x_i if the optimal value of x_i does not depend on the value any other design variable takes on. f is said to be separable if and only if it is separable with respect to all components of \vec{x} .

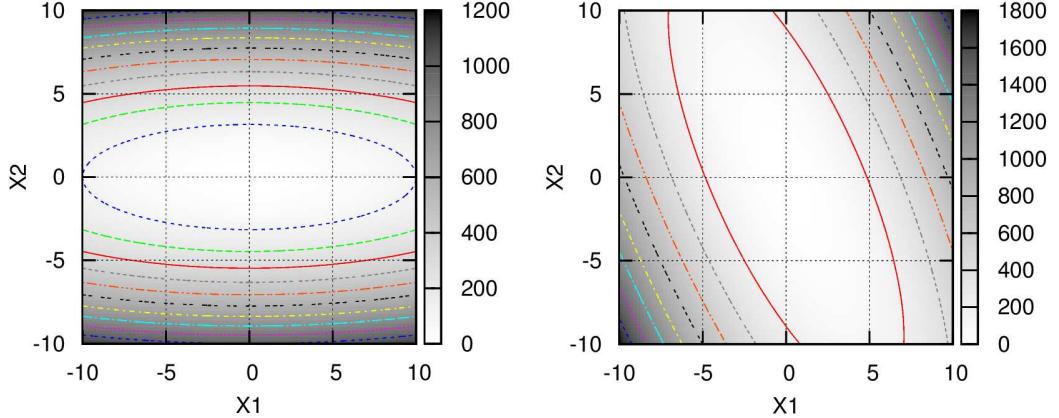


Figure 4.2: Separable (left) and non-separable (right) objective functions.

As mentioned above, an ill-posed optimization problem suffers from the curse of dimensionality, the computational cost to solve it scales superlinearly with the number of design variables N . To better understand this, let us first examine a separable objective function. The cost for solving an optimization problem with a separable objective function increases linearly with N , since the global optimum can be located by minimizing N objective functions with a single design variable each (separately minimise $f_i(x_i)$, $\forall x_i \in \vec{x}$). Thus, the cost of solving a separable N -dimensional optimization problem is approximately N times the cost of solving the equivalent 1D problem. On the other hand, this is not possible for an ill-posed optimization problem since, for at least one design variable, its optimal value depends on the values some or all of the other take on. Since the search space increases exponentially with N , this leads to a superlinear (with N) increase in the cost of solving ill-posed problems.

4.1.3 Investigation of EAs Efficiency Degradation in Ill-Posed Problems

In order to investigate the effects of ill-posed optimization problems on EA's efficiency, two mathematical test cases with different N values are solved, both in their separable (well-posed) and non-separable (ill-posed) form. The results presented in this section correspond to the mean performance of 10 runs, with different random number generator seeds.

The first analytical test case to be examined is a multi-dimensional ellipsoid (a 2D version of it can be seen in fig. 4.2) described, in its separable form, by

4. EAs and MAEAs Driven by Principal Component Analysis

$$f(\vec{x}) = \sum_{i=1}^N a^{\frac{i-1}{N-1}} x_i^2 \quad (4.2)$$

where the value of a determines the anisotropy of the objective function. Large values of a lead to increasingly anisotropic functions.

The same problem can be examined after casting it in non-separable form, as follows

$$f(\vec{y}) = \sum_{i=1}^N a^{\frac{i-1}{N-1}} y_i^2 \quad (4.3)$$

where $\vec{y} = B\vec{x}$ and B is an appropriate $N \times N$ orthogonal rotation matrix.

In order to investigate both the effects of the design space dimension (N) and the condition number (a), four different optimization problems were solved. A 10D ($N = 10$) optimization problem was solved for $a = 10$ and $a = 100$, fig. 4.3, and a $N = 30$ problem for $a = 100$ and $a = 1000$, fig. 4.4.

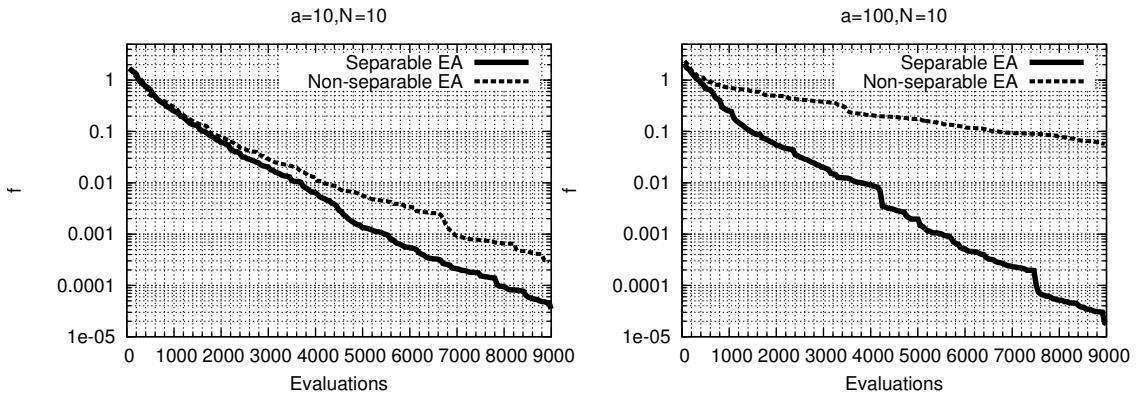


Figure 4.3: 10D ellipsoid. Condition number $a = 10$ (left). Condition number $a = 100$ (right). Increasing a from 10 to 100 increases the EAs performance gap between the separable and non-separable problems.

In figs. 4.3 and 4.4, one may identify both the effects of increasing the condition number and increasing the problem dimension. A condition number with higher value leads to greater performance gap between the separable and non-separable cases, for both $N = 10$ and $N = 30$ cases. Increasing the problem dimension has the same effect. Focusing on the $a = 100$ case, for both $N = 10$ and $N = 30$, one may observe that the difference in objective function values $|\Delta(f)|$, for the same computational cost of 9000 evaluations, differ. For the $N = 10$ case,

4.1 Ill-Posed Optimization Problems

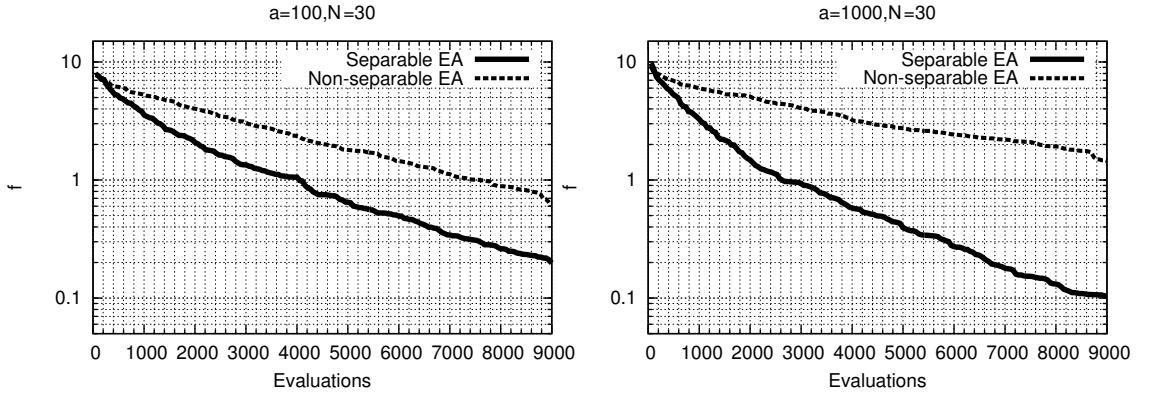


Figure 4.4: 30D ellipsoid. Condition number $a = 100$ (left). Condition number $a = 1000$ (right). As in the 10D case, increasing a from 100 to 1000 causes additional losses in performance and increases the performance gap between separable and non-separable optimization problems.

$|\Delta(f)| = |f(\text{non-separable}) - f(\text{separable})| \approx 10^{-4}$. On the other hand, for the $N = 30$ case $|\Delta(f)| \approx 0.9$. This difference in $|\Delta(f)|$, between the 10D and 30D case, denotes a significant increase in the efficiency gap between separable and non-separable cases when the dimension is increased.

The second analytical test case is concerned with the minimization of a multi-modal objective function described by

$$f(\vec{x}) = 10N + \left(\sum_{i=1}^N x_i \right)^2 - 10N \cos\left(\pi \sum_{i=1}^N x_i\right) \quad (4.4)$$

A 2D example is shown in fig. 4.5. As it appears in eq. 4.4, the problem is non-separable. It can, though, easily be transformed to a separable one if \vec{x} changes to $\vec{y} = B\vec{x}$, where B is an appropriate rotation matrix. A 45° rotation is employed in each direction. This objective function is extremely anisotropic since only one direction in the design space contributes to the objective function value and all the rest become irrelevant. Therefore, the condition number a equals practically to ∞ . This is an important class of objective functions since it resembles MOO problems as shown in section 4.2.

In order to investigate the effects of non-separability, as a function of the problems dimension, two optimization problems (5D and 30D) were solved.

In fig. 4.6, it is demonstrated that both problems, if reformulated using separable objective functions, outperform the non-separable ones. One may also observe that, the increase in problem's dimension causes an increase in efficiency gain, between separable and non-separable function cases. This conclusion is

4. EAs and MAEAs Driven by Principal Component Analysis

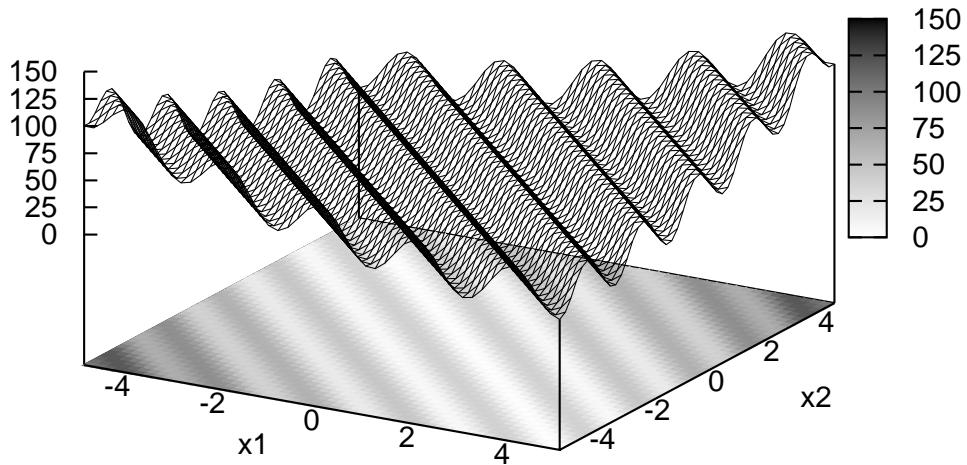


Figure 4.5: Minimization of eq. 4.4, for $N=2$. The optimal value of x_1 depends on the choice of x_2 and vice-versa. So, this objective function is non-separable in terms of x_1 and x_2 . Nevertheless, by aligning one of the two design variables with the $x_1=x_2$ axis, the problem becomes separable.

similar to the one made for the ellipsoid test case.

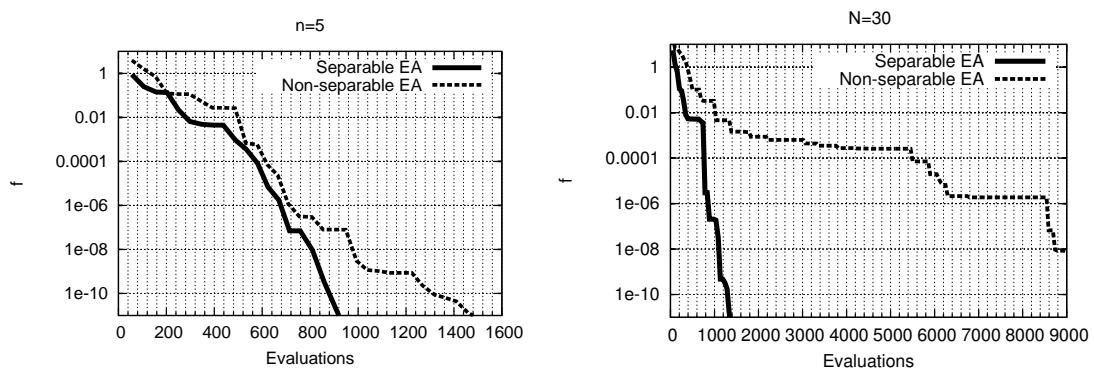


Figure 4.6: Solution of the 5D problem of eq.4.4 (left). Solution of the 30D problem of eq. 4.4 (right). The loss in efficiency for problems with extremely anisotropic objective functions increases significantly with the design space dimension.

Through the preformed investigations (figs. 4.3, 4.4 and 4.6), it is convincingly shown that a significant amount of the EA efficiency can be lost when dealing with ill-posed optimization problems, especially when these are combined with high values of N . The degree of efficiency deterioration is proportional to three quantities:

- a) The anisotropy magnitude: Increasing anisotropy leads to less efficient EAs (figs. 4.3 and 4.4).
- b) Function separability: Problems with separable objective functions with respect to their design variables can be solved significantly faster using EAs, as shown in figs. 4.3, 4.4 and 4.6.
- c) The number of design variables N : It is obvious that, by increasing the number of design variables, any EA requires more computational resources to locate the optimal solution. Additionally, if combined with (b), the great value of N leads to superlinear increase in cost (fig. 4.6).

4.2 Ill-Posed MOO Problems

In ill-posed MOO problems, all comments made and conclusions reached must now be applied to the scalar cost function Φ (rather than the objective function f), since Φ is the quantity which “drives” the evolution operators. In MOO, ill-posed optimization problem are typically associated with functions Φ which are non-separable and extremely anisotropic. This results from the fact that a Pareto front of optimal non-dominated solutions is sought. Since the Φ values given to the population members depend upon the Pareto dominance, all members of the current front of non-dominated solutions have $\Phi=0$ (or very small values compared to the dominated members of the population if niching effects are taken into account). These leads to directions in the design space that have very small impact on Φ , i.e. directions along the Pareto front ($\Phi \approx 0$) and directions with high impact on it, i.e those perpendicular to the Pareto front, therefore, almost always, resulting in an extremely anisotropic problem.

4.3 Principal Component Analysis

The aim is to find the appropriate transformation of the original set of variables $\vec{x} \rightarrow \vec{x}^*$, capable to transform the ill-posed optimization problem in hand into a better-posed one. After this transformation, the objective function in SOO or the scalar cost function Φ in MOO is expected to have enhanced separability

4. EAs and MAEAs Driven by Principal Component Analysis

properties with respect to the transformed design variables \vec{x}^* . This procedure is equivalent to finding an orthogonal matrix U ($UU^T = I$) and based on it, express \vec{x}^* as $\vec{x}^* = U\vec{x}$. In this thesis, it is proposed that the topological characteristics of the Pareto front determine the appropriate transformation matrix U . To extract these in a way which best represents the data variance, the Principal Component Analysis (PCA) method is used, (82, 91). The transformation matrix U contains the Principal Directions (PDs) computed by the PCA applied to the members of the Pareto front.

However, the Pareto front is the final goal of the optimization process and therefore, it is unknown during the optimization. Instead, in the proposed method, in each EA generation, the front of non-dominated solutions (elite set) is the one undergoing PCA. This is a reasonable choice since, as the evolution proceeds, the current elite-set tends to the Pareto front. Thus, the PDs computed by processing the elite-set in each generation become progressively more reliable approximations of the PDs computed by analysing the Pareto front.

Applying PCA to the current elite-set calls for the transformation of its members into a standardized data-set X , with $\mu = 0$ and $\sigma = 1$ in all directions. Next step is the computation of the empirical covariance matrix P , (50, 91),

$$P = \frac{1}{e} XX^T \quad (4.5)$$

where e is the size of the elite-set and X the matrix formed using the members of the elite-set as columns. Then, via the spectral decomposition theorem, (18, 50), P can be written as

$$P = U\Lambda U^T \quad (4.6)$$

where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$ is a diagonal matrix with the eigenvalues of P and U is a $N \times N$ matrix containing the eigenvectors (PDs) as columns.

4.4 The Welded Beam Case

To better demonstrate the ill-posed nature of MOO problems, a modified version of the welded beam case (28) is used, (fig. 4.7). The minimization of both the cost (K) and the deflection (Δ) of a welded beam subject to force P (fig. 4.7) is desired. There are two design variables: the welding length (x_1 or l) and the side length of the square cross section (x_2 or t). The design is subject to three constraints regarding shear stress (τ), bending stress (σ) and buckling load (P_c).

The problem requires the minimization of both the cost

$$K = 1.10471h^2l + 0.04811t^2(14.0 + l)$$

4.4 The Welded Beam Case

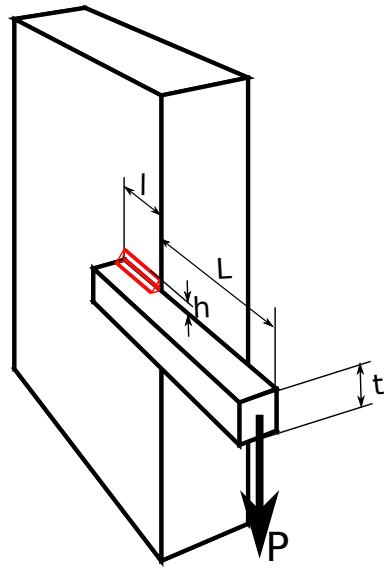


Figure 4.7: The Welded Beam Case: In the examined case, the welding length ($x_1=l$) and the square cross section side-length ($x_2=t$) are the two design variables.

and the deflection

$$\Delta = \frac{4PL^3}{Et^4} = \frac{4PL^3}{EX_2^4} \quad (4.7)$$

The first constraint is related to the shear stress applied to the welding in order to ensure its structural integrity. The maximum shear stress depends on the type and quality of the welding. This constraint is expressed as

$$\tau = \sqrt{\tau_1^2 + \frac{\tau_1\tau_2 l}{R} + \tau_2^2} \leq 13600 \text{ psi} \quad (4.8)$$

where

$$\tau_1 = \frac{P}{\sqrt{2}hl}, \quad \tau_2 = \frac{MR}{J}, \quad M = P(L + \frac{l}{2})$$

$$R = \sqrt{\frac{l^2}{4} + \left(\frac{h+t}{2}\right)^2}$$

$$J = 2 \left(\sqrt{2}hl \left(\frac{l^2}{12} + \left(\frac{h+t}{2} \right)^2 \right) \right)$$

4. EAs and MAEAs Driven by Principal Component Analysis

The second constraint is on the bending stress applied to the beam, in order to ensure its structural integrity. The maximum bending stress depends on the beam material,

$$\sigma = \frac{6PL}{t^3} \leq 30000 \text{psi} \quad (4.9)$$

Finally, in order to avoid buckling, the buckling load is also constrained as follows,

$$P_c = \frac{4.013E\sqrt{\frac{t^8}{36}}}{L^2} \left(1 - \frac{t}{2L}\sqrt{\frac{E}{4G}} \right) \geq P \quad (4.10)$$

where the welding height is $h = 0.6 \text{in}$, the beam length is $L = 14 \text{in}$, the force is $P = 6000 \text{lb}$, the Young's modulus is $E = 30 \times 10^6 \text{psi}$ and the shear modulus is $G = 12 \times 10^6 \text{psi}$.

Apart from the constraints which are related to structural integrity, two additional constraints, related to both objectives, were imposed. These are

$$K \leq 50 \quad (4.11)$$

$$\Delta \leq 0.25 \quad (4.12)$$

From the Φ plot shown in fig. 4.8 (left), it is evident that the problem in hand is both extremely anisotropic and non-separable (ill-posed). Using the proposed PCA of the elite-set (fig. 4.8, right) the computed PDs point to the directions in the design space which if used to define new design variables, would transform the ill-posed problem into a better-posed one. The direction with the largest eigenvalue λ_1 is the direction that describes the Pareto front is aligned with the $\Phi = 0$ curve. On the other hand, the PD with the smallest eigenvalue λ_2 is the direction in the design space which corresponds to the largest change in Φ values. Smaller eigenvalue means smaller variance in the corresponding direction, in the elite-set.

4.5 PCA-Driven Evolution Operators

In EA-based optimization, the application of the evolution operators may benefit a lot from the outcome of PCA, in order to reduce the computational cost required for the same quality of solutions. To be more specific, PCA is used to identify directions in the design space which, if used to define new optimization variables, would lead to an optimization problem with (almost) separable objective function. The design space is, then, temporarily aligned with the principal directions and

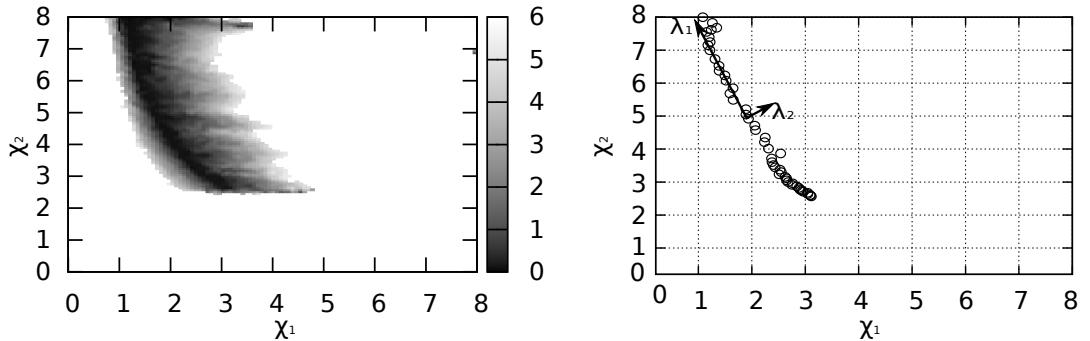


Figure 4.8: The Welded Beam Case: Left; Φ iso-areas over the design space, at a specific generation during the evolution. Right; The elite-set, plotted in the design space. Based on it, two PDs denoted by λ_1 and λ_2 are computed.

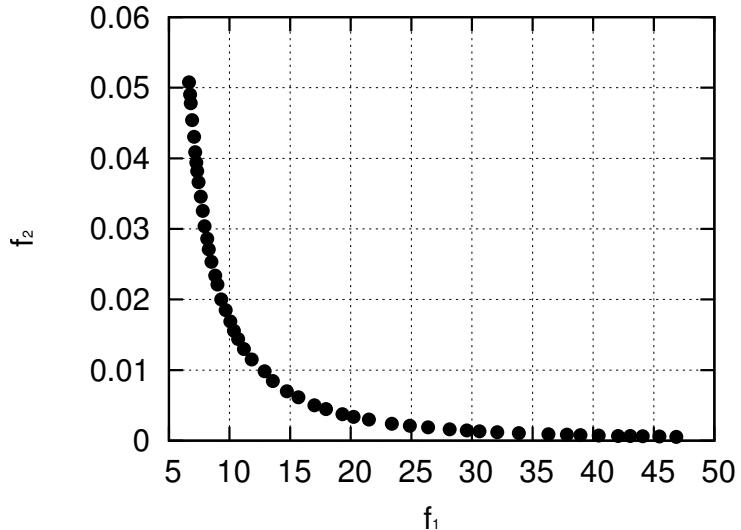


Figure 4.9: The Welded Beam Case: The elite-set shown in fig.4.8 is plotted in the objective space, where f_1 = cost and f_2 = deflection.

the evolution operators apply to individuals expressed in the so-rotated axes. At the end of this phase, the generated offspring are transformed back to the original axes of the design space. The rotation of the design space axes occurs just before and after the application of the evolution operators thus deeming it transparent to the designer. To perform the PCA, a small number of well-performing solutions to the problem must be available. Herein, these are the members of the current elite-set, as stated before; this set is dynamically updated during the evolution. In MOO applications, the elite-set comprises the members of the current front of non-dominated solutions. In SOO problems, where the notion of dominality degenerates, the PDs can be found by processing a user-defined number of the

4. EAs and MAEAs Driven by Principal Component Analysis

current top individuals.

Let the aligned with the PDs design vectors \vec{x}_i be represented by \vec{x}_i^* . The alignment is performed as follows

$$\vec{x}_i^* = U(\vec{x}_i - \vec{\mu}_X) \quad (4.13)$$

where $\vec{\mu}_X$ is the vector of mean (over the elite set) design variables. The inverse transformation, from \vec{x}_i^* to \vec{x}_i , is given by

$$\vec{x}_i = U^{-1}\vec{x}_i^* + \vec{\mu}_X \quad (4.14)$$

It should be stated that the CPU cost to compute U^{-1} is negligible since U is an orthogonal matrix and, thus, $U^{-1} = U^T$.

The two main PCA-driven evolution operators, recombination and mutation, are presented below.

4.5.1 PCA-Driven Recombination

The recombination operator is applied to the rotated design variable-set which are sought to be as separable as possible. As shown in fig. 4.10, this changes the probability of the offspring appearance in the design space as if the problem was separable (fig. 4.8).

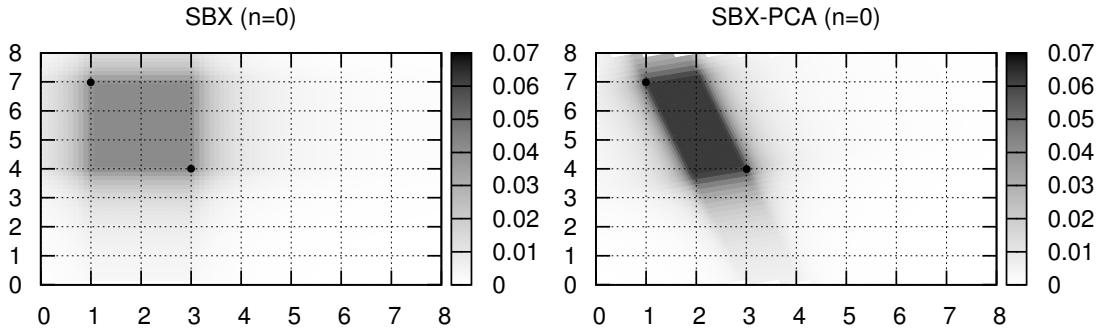


Figure 4.10: The Welded Beam Case: The Offspring appearance probability using the SBX($n=0$) recombination scheme (section 2.3.1.1). Assume that there are two parents shown by the two black dots. Left; SBX recombination. Right; PCA driven SBX recombination.

From figs. 4.8 and 4.10, it can be seen that the proposed method assigns higher offspring appearance probability to the regions near $\Phi = 0$ which is the region that accommodates the currently best (elite) individuals. Therefore, the

PCA-driven recombination better achieves its goal of mixing, according to the building block hypothesis (76). Mixing is the process in which building blocks (low-order schemata of high fitness) are combined to form new potentially fit schemata of higher order. The proposed method increases the probability individuals generated through the application of the recombination operator to be fit.

4.5.2 PCA-Driven Mutation

Mutation is also applied to the design space aligned with the computed PDs. The mutation probability on each PD is dynamically updated based on the variance (\vec{V}) of the elite population members, projected onto the aforementioned direction ($V(i)$), eq. 4.15. This is done in a way that keeps the overall mutation probability per individual constant (and equal to a user-defined value). The proposed method distributes the mutation probability among the PDs in a way which is proportional to the importance of each direction, according to the characteristics of the current front of non-dominated solutions. As shown in the welded beam case, the importance of each PD, regarding either f (SOO) or Φ (MOO), is inversely proportional to its eigenvalue which quantifies the variance. PDs with high-valued eigenvalues receive smaller mutation probability, in contrast to the directions with low-valued eigenvalues which undergo mutation with higher probability. The formula used to adjust the mutation probability (per each newly computed direction (index i) is

$$p_m^i = 0.1p_m + \frac{0.9p_m N}{D_V} \cdot \frac{V_{max} - V_i}{V_{max} - V_{min}} \quad (4.15)$$

where p_m is a user-defined mutation probability, $V_{max} = \max\{V_1, \dots, V_N\}$, $V_{min} = \min\{V_1, \dots, V_N\}$ and

$$D_V = \sum_{i=1}^N \frac{V_{max} - V_i}{V_{max} - V_{min}} \quad (4.16)$$

Let us denote by g the generation counter, by k_{DB} the number of currently existing entries into the DB and by k_{min} the minimum number of entries needed to initiate the use of the metamodel-based pre-evaluation phase. Then, the successive steps of the PCA-driven MAEA or MAEA(PCA) can be described as follows:

Step 1: For the current offspring population S_λ^g , set $\lambda^* = \lambda$ (if $k_{DB} < k_{min}$) or $\lambda^* = \lambda_e$ (else), where λ_e the number of offspring to be evaluated on the problem-specific evaluation software.

4. EAs and MAEAs Driven by Principal Component Analysis

Step 2: If $k_{DB} \geq k_{min}$, λ RBF networks are trained on paired input–output patterns selected from the DB, in the vicinity of each offspring. Pre-evaluate the λ population members on the so-trained metamodels. Select the λ_{ex} top of them, based on dominance and proximity criteria.

Step 3: Evaluate the λ_e offspring on the problem-specific evaluation software. Update S_e^g .

Step 4: Compute the current set of PDs based on the updated elite-set S_e^g (eqs. 4.5 and 4.6).

Step 5: Align the design space with the so-computed PDs (eq. 4.13).

Step 6: Create the new offspring population S_λ^{g+1} through the application of the evolution operators to the aligned individuals. Re-align the generated offspring with the original design space coordinate system (eq. 4.14).

Step 7: Set $g \leftarrow g + 1$; go to Step 1.

The above algorithm can readily be transformed to EA(PCA) by eliminating the use of metamodels or by setting k_{min} to an “infinite” number.

4.5.3 Gain from the Use of EA(PCA)

In order to demonstrate the performance gain from the use of the EA(PCA), the welded beam case and the two mathematical minimization problems already presented in section 4.1.3 are revisited. For each one of them, 10 runs were performed with different random number generator seeding.

Regarding the two-objective welded beam problem, the average hypervolume indicator plotted in terms of the number of evaluations performed is presented in fig. 4.11. In this figure, it is shown that, even though the dimension of the problem was deliberately kept low ($N = 2$) to facilitate the visual analysis of the obtained results, the gain from the use of the proposed method over the conventional EA is clear.

The comparison between conventional evolution operators and the PCA-driven ones, on the 30D non-separable ellipsoid with $a = 1000$ (eq. 4.3), is presented in fig. 4.12. This figure presents the mean objective function values of 10 runs.

In fig. 4.12, the significant gain from the use of the proposed EA(PCA) in comparison with an EA in the 30D ellipsoid case is shown.

Regarding the multi-dimensional and multi-modal problem of eq. 4.4 with $N=30$, a comparison between PCA-driven and conventional evolution operators is presented in fig. 4.13. In this figure, it is shown that the use of PCA-driven evolution operators significantly enhances the EA efficiency in the 30D multi-modal problem, too.

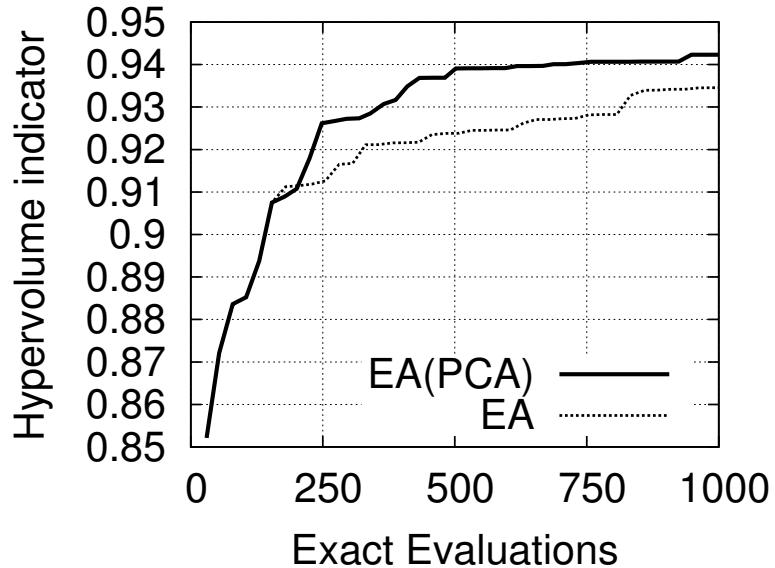


Figure 4.11: The Welded Beam Case: Average evolution of the hypervolume indicator using EA and EA(PCA). In this case, metamodels are not used due to the extremely small computational cost of each evaluation run. EA(PCA) outperforms traditional EA even though the dimensionality of the problem is very low.

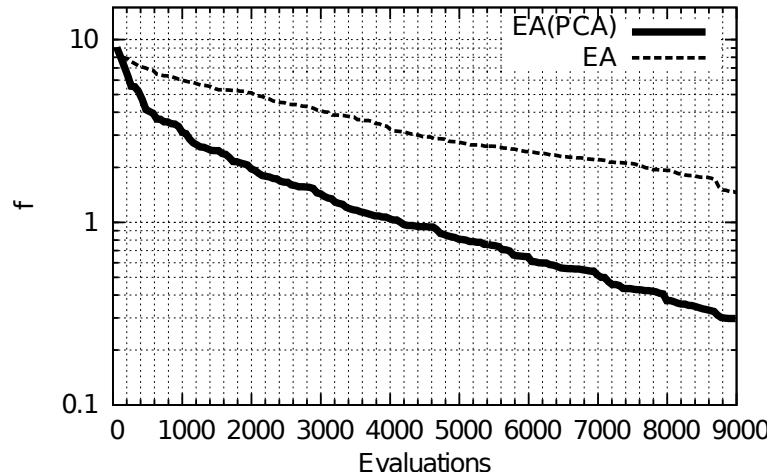


Figure 4.12: 30D non-separable ellipsoid with $a=1000$: Average convergence using EA and EA(PCA). The proposed PCA-driven evolution operators significantly outperform the conventional ones.

4.6 EAs Assisted by PCA-Driven Metamodels

In MAEAs, the main challenge regarding the use of artificial neural networks as metamodels is to overcome the so-called curse of dimensionality in problems with

4. EAs and MAEAs Driven by Principal Component Analysis

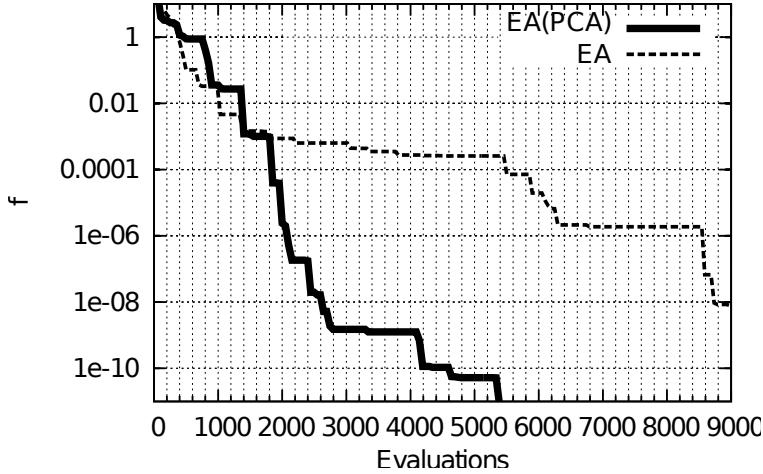


Figure 4.13: 30D multi-modal problem: Average convergence using the EA and the EA(PCA). Similar to fig. 4.12, the proposed PCA-driven evolution operators significantly outperform the traditional ones.

a large number of design variables. The number of training patterns required to build a reliable metamodel increases with N ; a side-effect is that their training becomes computationally costly. On the other hand, the need for more training patterns per metamodel delays the commencement of the IPE phase during the MAEA, until the DB collects a substantial number of entries. This leads to a further increase in the wall clock time of the optimization run.

This thesis proposes the reduction in the number of the RBF network sensory (input) nodes and, consequently, the required number of training patterns. The eigenvectors computed through the PCA of the elite-set are associated with the objective function variance. If the elite-set is expressed in a coordinate system aligned with the PDs, it is expected that any plot of the objective function (SOO) or Φ (MOO) in terms of the first components (i.e. those with the highest variance) must be scattered enough. In contrast, along the PDs with small variances, the elite members are much less scattered.

The proposed method takes advantage of the above and cuts off a pre-selected number of the RBF network sensory units, particularly those related to the highest variance directions in the design space. As a result, the RBF networks are trained on lower dimension data. This truncation is applied only during the RBF network training, by firstly rotating/aligning all the selected training patterns with the computed PDs and, then, cutting those sensory nodes which correspond to the highest eigenvalues off. The number of the components to be cut off is user-defined.

Fig. 4.14 presents an instant of the evolution process, at the end of the 6th

4.6 EAs Assisted by PCA-Driven Metamodels

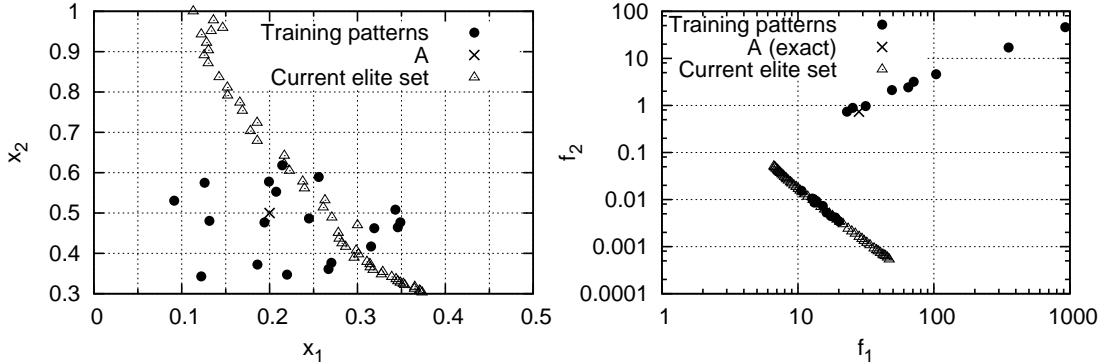


Figure 4.14: The Welded Beam Case: An instant of the evolution process, where the current elite set, a candidate solution (A) and the training patterns selected from the current DB for training the metamodel to be used to predict the objective function value at A are plotted in the design (left) and objective (right) space.

generation of one of the runs. At this instant, λ new offspring have just been generated. Let us assume that the candidate solution A, with $(x_1, x_2) = (0.2, 0.5)$ shown in fig. 4.14, needs to be pre-evaluated using an appropriately trained local RBF network. For this purpose, 19 patterns are selected from the DB using the standard selection procedure. In fig. 4.14, the training patterns, the candidate solution and, also, the current elite-set are plotted both in the design (x_1, x_2) and objective (f_1, f_2) spaces. The exact values of f_1 and f_2 for A are also shown. It is evident that, in a problem with only two design variables, there is no reason to reduce the number of design variables during the training of the RBF networks. However, it was decided to do so, just for the purpose of demonstration.

A first demonstration of the proposed method is based on the welded beam problem (fig. 4.7). In fig. 4.15, the f_1 predictions using two RBF networks, with either x_1 or x_2 as the only sensory unit of each network, are shown. Though these networks, after being trained on the selected 19 patterns, must be used to predict the objective functions only at A, it is interesting to observe the overall deviation between the continuous line which stands for the network response and the training patterns' responses. Recall that the number of RBF centers is less than the number of training parameters, so the network does not interpolate the training patterns. The observed high deviations, which could be even higher for some other candidate solutions, are easily explained by the fact that a two-variable function $f_1(x_1, x_2)$ is handled as $f_1(x_1)$ or $f_1(x_2)$. The corresponding results for f_2 , instead of f_1 , are shown in fig. 4.16. The exact objective function values at A along with the RBF predictions are tabulated in table 4.1. It can be seen that f_1 is predicted with a relative error of $\approx 57\%$ if $f_1 = f_1(x_1)$ and $\approx 138\%$

4. EAs and MAEAs Driven by Principal Component Analysis

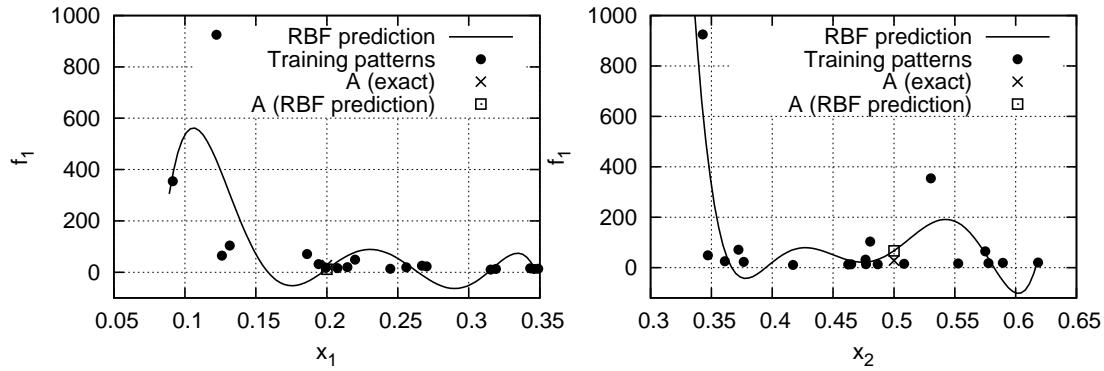


Figure 4.15: The Welded Beam Case: RBF network predictions of f_1 if x_1 (left) and x_2 (right) are separately applied to the single sensory unit of the network.

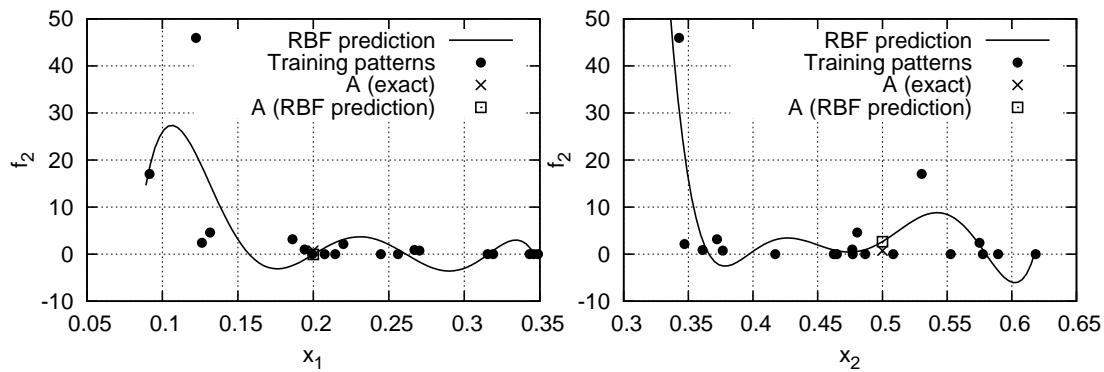


Figure 4.16: The Welded Beam Case: RBF network predictions of f_2 if x_1 (left) and x_2 (right) are separately applied to the single sensory unit of the network.

4.6 EAs Assisted by PCA-Driven Metamodels

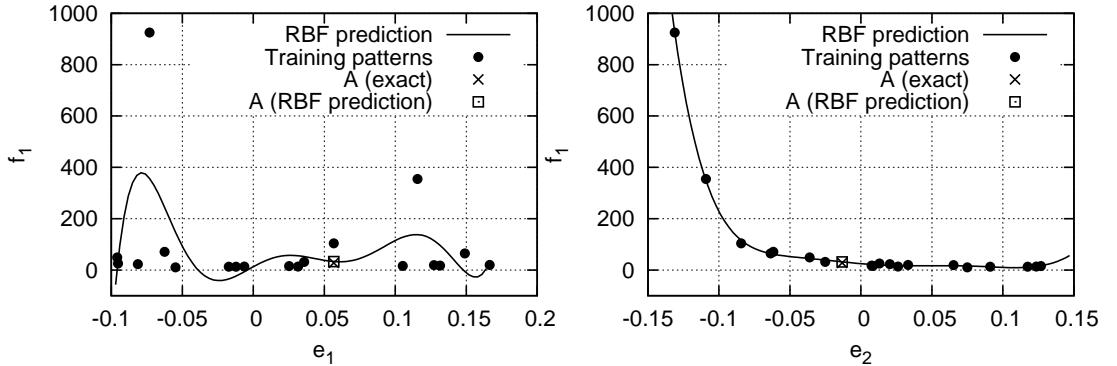


Figure 4.17: The Welded Beam Case: RBF network predictions of f_1 if e_1 (left) and e_2 (right) are separately applied to the single sensory unit of the network.

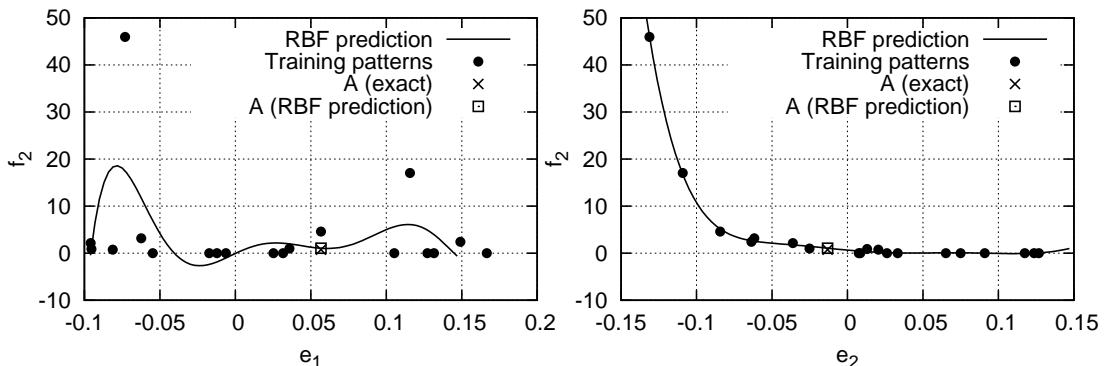


Figure 4.18: The Welded Beam Case: RBF network predictions of f_2 if e_1 (left) and e_2 (right) are separately applied to the single sensory unit of the network.

if $f_1 = f_1(x_2)$ and the error of f_2 is $\approx 110\%$ if $f_2 = f_2(x_1)$ and $\approx 260\%$ if $f_2 = f_2(x_2)$.

Exact (f_1, f_2) values	RBF network predictions assuming $f_i = f_i(x_1)$	RBF network predictions assuming $f_i = f_i(x_2)$
f_1 28.07	11.88	66.9
f_2 0.73	-0.08	2.66

Table 4.1: The Welded Beam Case: Exact response and RBF network predictions of f_1 and f_2 at A, using either x_1 or x_2 as the only input to the network.

The PCA of the current elite-set is used to compute the two PDs which, in turn, are used to select the single variable that must preferably be associated

4. EAs and MAEAs Driven by Principal Component Analysis

with the only sensory unit of the RBF network. The design space is transformed according to the computed PDs and the RBF network training is repeated using either e_1 or e_2 , instead of x_1 and x_2 , as input. The RBF networks' predictions are shown in fig. 4.17 and 4.18 and table 4.2. In particular, f_1 is predicted with a relative error of $\approx 17\%$ if $f_1 = f_1(e_1)$ and $\approx 13\%$ if $f_1 = f_1(e_2)$ and f_2 with $\approx 41\%$ if $f_2 = f_2(e_1)$ and $\approx 43\%$ if $f_2 = f_2(e_2)$. From these figures, it is concluded that the RBF networks can be trained only on e_2 , i.e. the PD with the smaller variance, since the training patterns are less scattered both in (e_2, f_1) and (e_2, f_2) and the obtained prediction is more dependable.

Exact (f_1, f_2) values	RBF network predictions assuming $f_i = f_i(e_1)$	RBF network predictions assuming $f_i = f_i(e_2)$
f_1 28.07	32.86	31.84
f_2 0.73	1.04	1.06

Table 4.2: The Welded Beam Case: Exact responses and RBF network predictions of f_1 and f_2 at A, using either e_1 or e_2 as the only input to the network.

4.6.1 Gain from the Use of M(PCA)AEA(PCA)

To demonstrate the gain in performance from the use of the proposed PCA-assisted metamodels, 10 runs using the so-called M(PCA)AEA(PCA) optimization for the two mathematical test cases presented in section 4.1.3 were performed with different random number generator seeding.

The comparison between runs based on the use of conventional metamodels where PCA is used only to drive the evolution operators (denoted by MAEA(PCA)) and additionally enhanced with PCA-assisted metamodels or M(PCA)AEA(PCA), for the 30D non-separable ellipsoid function with $a = 1000$ (eq. 4.3), is presented in fig. 4.19. The plots refer to the mean objective function values of 10 runs. The IPE phase for the MAEA(PCA) initiated after 300 individuals were stored in the DB and 25 to 29 training patterns, in the 30D space, were used to train the RBF networks employed as metamodels. Regarding the M(PCA)AEA(PCA), the IPE phase started after only the first 100 individuals were stored in the DB since only 4 to 8 training patterns, in the truncated 10D space, were required for the RBF network training. In the latter case, the RBF networks were trained on the 10 most important directions in the design space, as identified by the PCA.

Regarding the multi-dimensional and multi-modal test case of eq.4.4, a comparison regarding EA(PCA), MAEA(PCA) and M(PCA)AEA(PCA) is presented, for the 30D problem, in fig.4.20. Regarding the MAEA(PCA), the IPE phase initiated after 300 individuals were stored in the DB and 15 to 19 training patterns in

4.6 EAs Assisted by PCA-Driven Metamodels

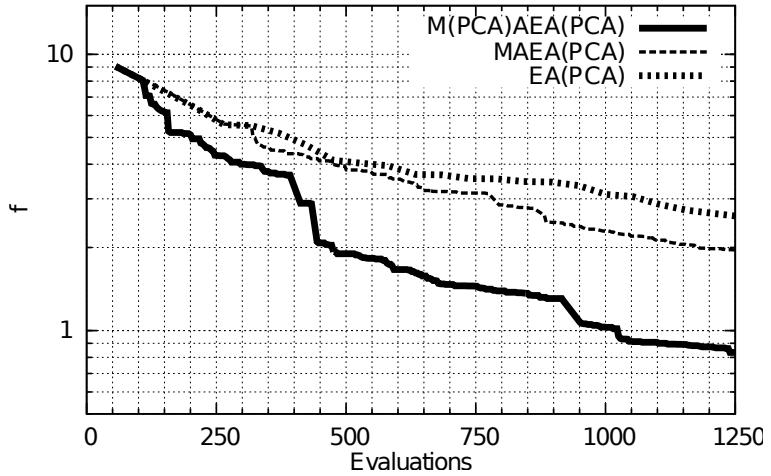


Figure 4.19: 30D non-separable ellipsoid with $a=1000$: Average convergence of EA(PCA), MAEA(PCA) and M(PCA)AEA(PCA). The proposed M(PCA)AEA(PCA) outperforms both of the other variants.

the \mathbb{R}^{30} space were used to train the metamodel. Regarding M(PCA)AEA(PCA), the training pattern dimension was truncated from 30 to the 10 most important ones with respect to f . This allowed for the IPE phase to start just after only 100 individuals were stored in the DB, since only 5 to 9 training patterns were required for the RBF network training.

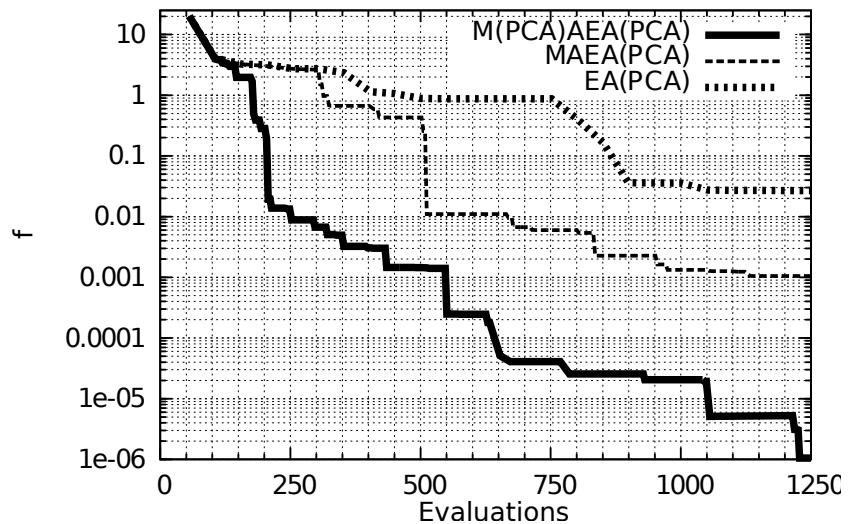


Figure 4.20: 30D multi-modal problem: Average convergence of EA(PCA), MAEA(PCA) and M(PCA)AEA(PCA). The proposed M(PCA)AEA(PCA) method performs much better than the other variants.

4. EAs and MAEAs Driven by Principal Component Analysis

In figs. 4.19 and 4.20, one may observe that the use of PCA-driven metamodels further enhances the MAEA efficiency.

4.7 Design of a Compressor Cascade

The design of a 2D compressor cascade operating at $M_1 = 0.54$, $a_1 = 44^\circ$ and $Re = 4 \times 10^5$ for minimum total pressure losses coefficient ω (eq. 3.7) is sought.

The blade airfoil is designed subject to a number of aerodynamic and geometrical constraints: the optimal airfoil must turn the flow by more than 30° and the blade airfoil thickness at three chord-wise positions $0.3c$, $0.6c$ & $0.9c$ must be greater than $0.10c$, $0.08c$ & $0.01c$, respectively.

The airfoil shape is parameterized based on its mean-camber line and superimposed thickness distributions, presented in section 3.3, yielding 27 design variables.

The advantages of using the M(PCA)AEA(PCA) method can be seen by comparing the performances of a conventional MAEA and the M(PCA)AEA(PCA). In both cases, the populations are $\mu = 20$, $\lambda = 60$ and $\lambda_e = 6$. Both cases used local RBF networks as metamodels trained on a small number of automatically selected patterns. The MAEA used a number of 20 to 30 training patterns in the $27D$ space. The IPE phase for the MAEA started after 400 individuals were stored in the DB. Regarding M(PCA)AEA(PCA), the RBF networks used as metamodels were trained on 5 to 8 patterns in the $10D$ space. The reduced dimension made the earlier use of metamodels possible; in the M(PCA)AEA(PCA) run, the IPE phase started after the first 200 individuals were stored in the DB.

Fig. 4.22 shows the convergence of both methods. M(PCA)AEA(PCA) outperforms the conventional MAEA, during all but the early generations of the optimization procedure.

The optimal airfoil resulting from M(PCA)AEA(PCA) run, shown in fig. 4.22, has a total pressure loss coefficient value of $\omega = 0.01803$ and respects both the thickness and flow turning constraints. The flow turning of the optimal blade is $\Delta a = 30^\circ$.

4.7 Design of a Compressor Cascade

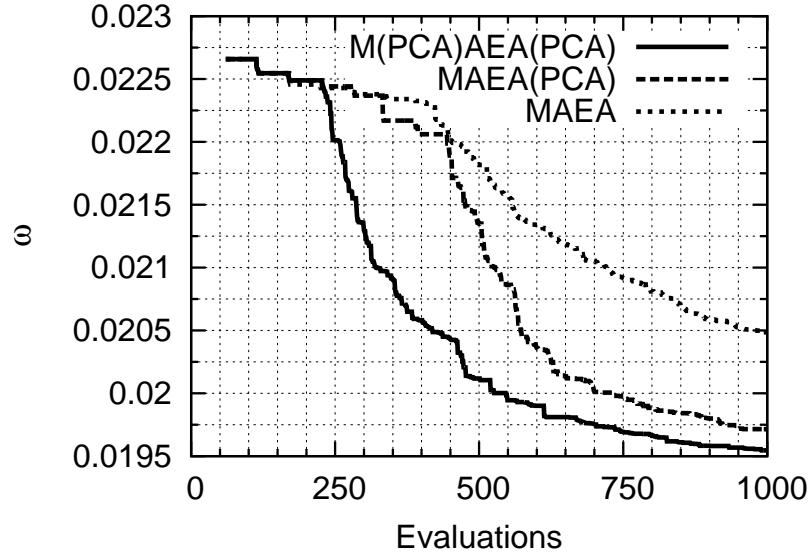


Figure 4.21: Design of a compressor cascade: Comparison of the mean value and standard deviation from 10 independent runs performed using MAEA, MAEA(PCA) and M(PCA)AEA(PCA).

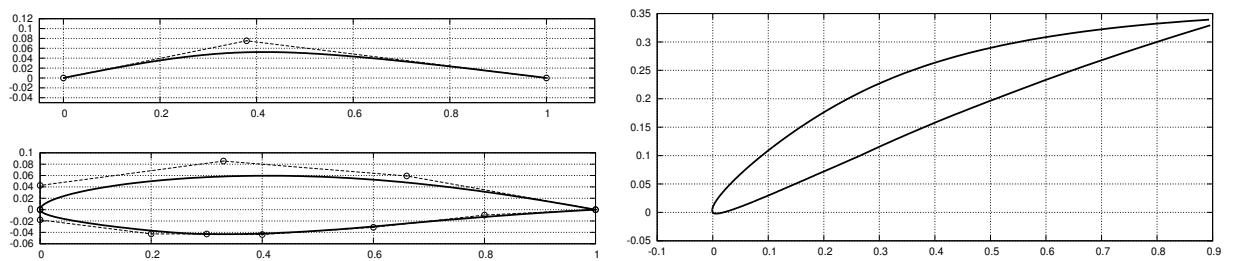


Figure 4.22: Design of a compressor cascade: The optimal airfoil resulted from M(PCA)AEA(PCA). Left: mean-camber line and thickness distribution together with their control polygons. Right: The final airfoil after superimposing thickness distributions on the mean camber line and turned to the fixed stagger angle.

4. EAs and MAEAs Driven by Principal Component Analysis

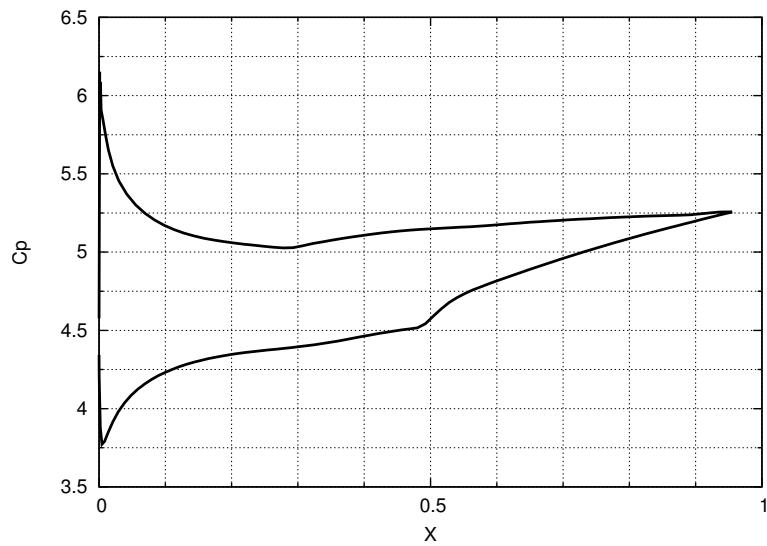


Figure 4.23: Design of a compressor cascade: Pressure coefficient C_p of the optimal airfoil shown in fig. 4.22.

5

Optimization of Hydraulic Turbomachines

In this chapter, the KBD method presented in Chapter 3 and the MAEA which is further driven by the principal component analysis (PCA) of promising individuals, presented in Chapter 4, are used to solve large scale industrial problems related to the design-optimizations of hydraulic turbines.

In order to perform a successful optimization, a reliable evaluation procedure must be available. The evaluation procedure used in this chapter (fig. 5.1) comprises three parts: the parameterization tool which is able to generate 3D turbine blade shapes, the grid generation tool able to automatically generate computational grids in any geometry created according to the aforementioned parameterization tool and the CFD software. Section 5.1 presents the basic features of the aforementioned tools. Section 5.1.1 is about the parameterization which is appropriate for 3D geometries of any type of reaction turbines. The grid generation is presented in section 5.1.2. And, finally, the Euler solver used to predict the water flow through the flow passage is presented in section 5.1.3.

Post-processing is an indispensable part of the evaluation procedure. Post-processing tools are used to compute the metrics associated with the fitness of each candidate solution, as presented in section 5.2.

Two problems are examined. The first problem is dealing with the design-optimization of a Francis turbine runner to be installed in a pre-existing hydroelectric plant. This case is mainly used to study the gain from the use of the developed KBD method (Chapter 3), as compared with the conventional EA, in an industrial environment where archived past designs, i.e. the outcomes of more or less similar projects, are, in fact, available.

The second problem is related to the design of a new type of hydraulic turbines, the so-called Hydromatrix®. This case is used to demonstrate the gain from the implementation of the PCA-driven evolution operators (MAEA(PCA)), as

5. Optimization of Hydraulic Turbomachines

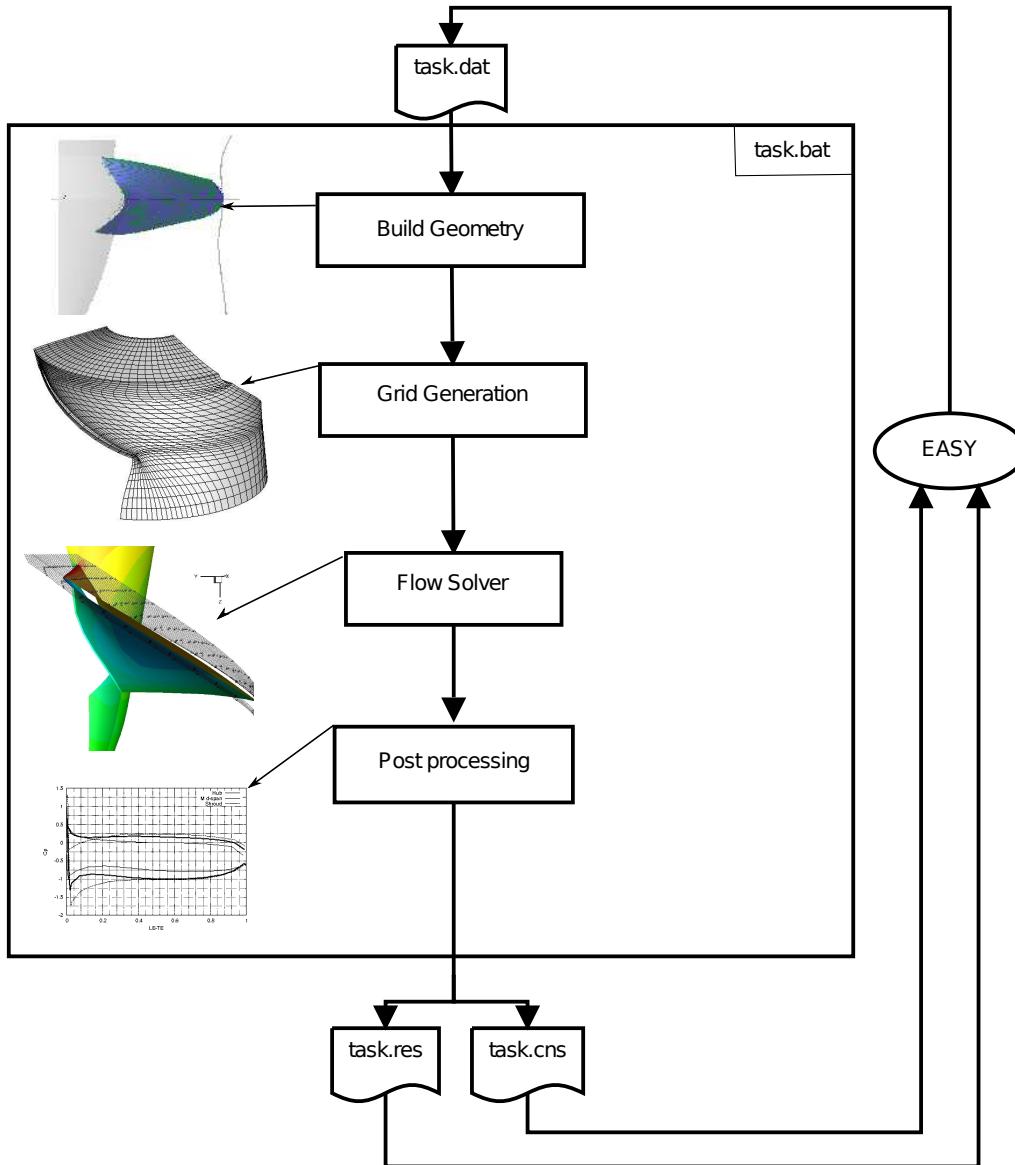


Figure 5.1: According to the EASY nomenclature, the evaluation of each new candidate solution created during the evolution starts by reading the corresponding values of the design variables from the “task.dat” file. A series of codes, being I/O compatible, are sequentially used to, finally, create the “task.res” and “task.cns” files which include the objective and constraint function values, respectively. The evaluation process is considered to be terminated once the “task.res” and “task.cns” files become available to the search engine of EASY. The names of the executable files required for a single evaluation are listed in the script file ”task.bat”.

compared with standard MAEA.

5.1 Evaluation Procedure

5.1.1 Parameterization for Turbine Blades

The parameterization of a hydraulic turbine blade, of axial radial or mixed flow type, consists of two steps. The first step is the parameterization of its 3D mean-camber surface. The second step is concerned with the thickness of the blade which is defined by the airfoil shape and the thickness distribution across the blade, (109, 182).

Starting point of the parameterization of the mean-camber surface is its meridional projection. The meridional projection of the mean-camber surface is confined by 4 curves, namely the hub and shroud generatrices, the leading and trailing edge curves (fig. 5.2). All four curves are parameterized by a user-defined number of Bézier control points.

Having defined the meridional projection of the mean-camber surface, the generation of the 3D shape of the mean-camber surface follows. A set of 2D iso-span lines, distributed between the shroud and hub, are generated (fig. 5.2). The 3D shape of the aforementioned lines is based on their meridional projection and a number of parameters, namely ρ , θ , β , ζ and μ , which are all defined below. These 3D lines are, then, used as the skeleton of the blade mean-camber surface (fig. 5.3).

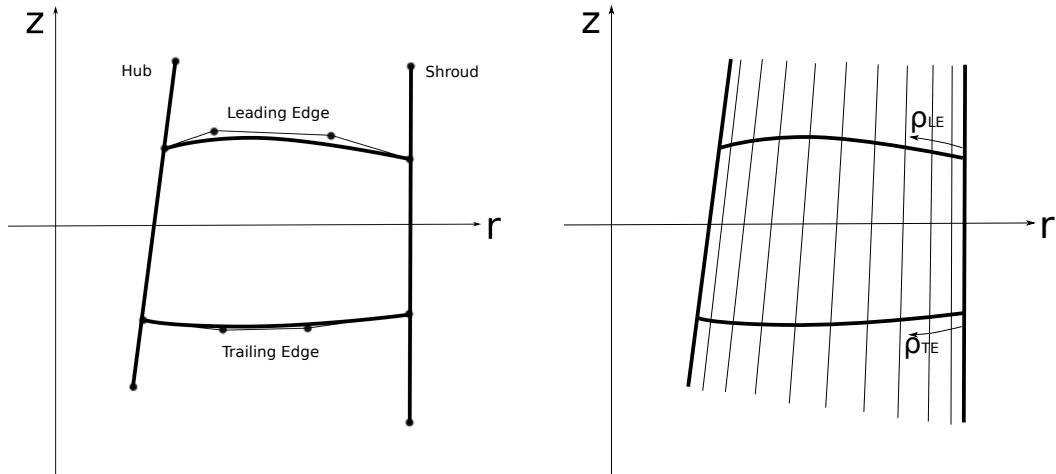


Figure 5.2: Left: Meridional projection of the mean-camber surface of a Hydromatrix® blade. Right: The 2D iso-span lines distributed between shroud and hub. Definition of ρ for the leading and trailing edge.

5. Optimization of Hydraulic Turbomachines

The ρ value is defined at each point along both the leading (LE) and trailing (TE) edges as the normalized meridional projection of the arc-length of the edge, with zero value at shroud and unit value at hub (fig. 5.2).

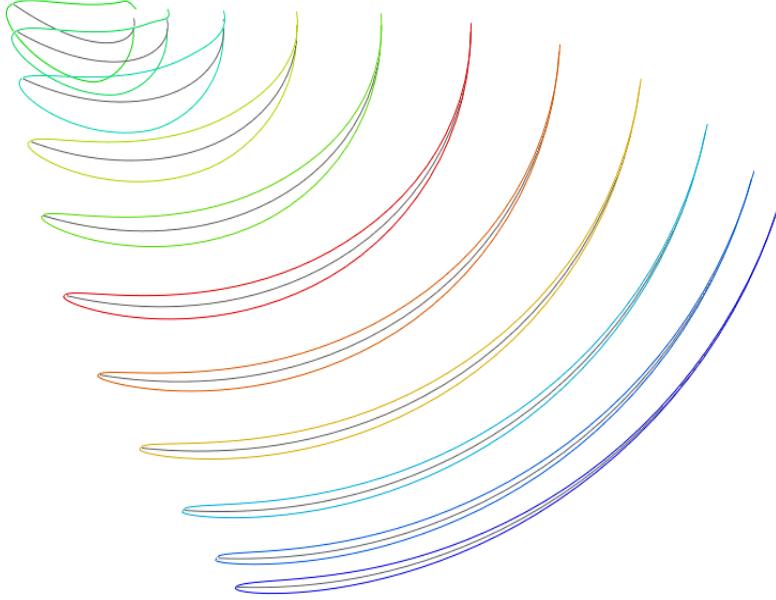


Figure 5.3: The 3D iso-span lines and their superimposed thickness profiles.

The angular position of the LE and TE is controlled by the so-called wrap-around angle θ . $\theta(\rho)$ distributions for the LE and TE together with the r and z values of their meridional projections define the final 3D shape of both edges. On the other hand, β is defined as the angle between the local tangent to the mean-camber surface and the tangent to the z -centered circle through this point. Therefore, the $\beta(\rho)$ distributions define the mean-camber surface metal angles throughout the previously defined LE and TE. $\beta(\rho)$ and $\theta(\rho)$ distributions are parameterized as Bézier curves of a user-defined degree (fig. 5.4).

The final step in the parameterization of the mean-camber surface is to define its shape between the LE and TE for all the iso-span lines. This is achieved through the conformal mapping Φ

$$\Phi : (r, z) \rightarrow \mu, \quad \mu = \int \frac{1}{r} dl \quad (5.1)$$

where l denotes the arc-length of the meridional projection of the iso-span line. This conformal mapping performs the angle-preserving transformation of the iso-span lines from the cylindrical (r, z, θ) to the (μ, θ) coordinate system. The iso-

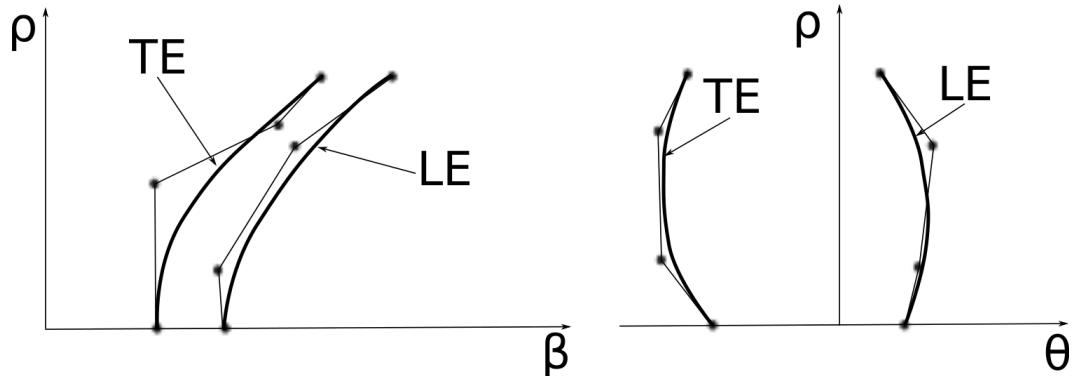


Figure 5.4: Left: $\beta(\rho)$ distribution. Right: $\theta(\rho)$ distribution.

span lines (in the (μ, θ) coordinate system) are defined by third-degree Bézier curves. The first and last control points are fixed on the LE and TE points of each iso-span line and the two internal control points depend on ζ_{LE} , ζ_{TE} , β_{LE} and β_{TE} angles, respectively (fig. 5.5).

Finally, the two $\zeta(\rho)$ distributions, for the LE (ζ_{LE}) and TE (ζ_{TE}), define the position of the two internal control points for all the iso-span lines ($0 \leq \rho \leq 1$). Therefore, the ζ distributions can be seen as the curvature control system of the aforementioned parameterization of the mean-camber surface scheme, since it defines how strongly the β values at LE and TE affect the β values in the interior of the mean-camber surface. Similar to $\theta(\rho)$ and $\beta(\rho)$, the $\zeta(\rho)$ distributions are also parameterized as Bézier curves of a user-defined degree (fig. 5.5). Then the inverse conformal mapping Φ^{-1} is used to transform the iso-span lines from the (μ, θ) coordinate system back to the cylindrical one.

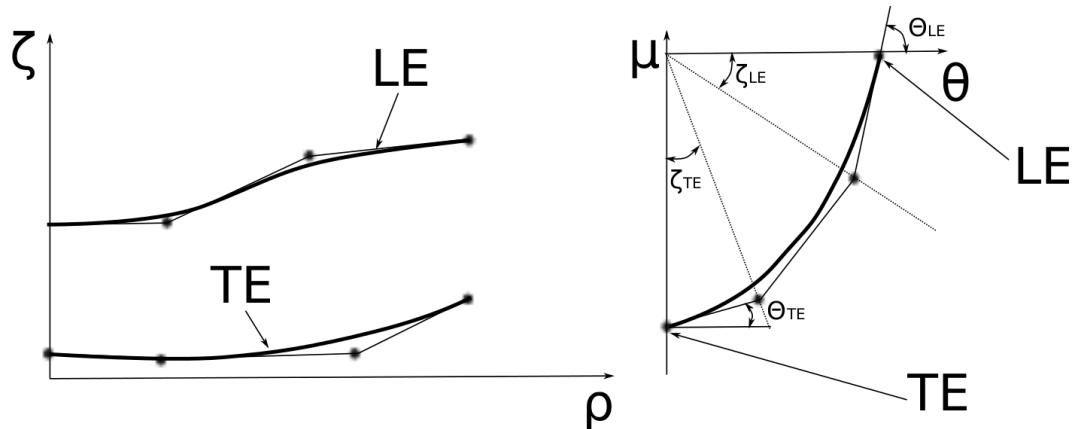


Figure 5.5: Left: $\zeta(\rho)$ distributions for ζ_{LE} and ζ_{TE} . Right: Iso-span line, in the (μ, θ) coordinate system, and its Bézier control point polygon.

5. Optimization of Hydraulic Turbomachines

The airfoil profiles are defined by a user-defined number of Bézier points (fig. 5.6) and the absolute thickness by two thickness ($t(\rho)$) distributions (fig. 5.7) (separately for the SS and PS). The airfoil profiles are scaled so that their maximum thickness be equal to the one specified by the $t(\rho)$ distribution. Finally, the scaled airfoil profiles are superimposed onto the mean-camber surface to create the final 3D blade (fig. 5.7).

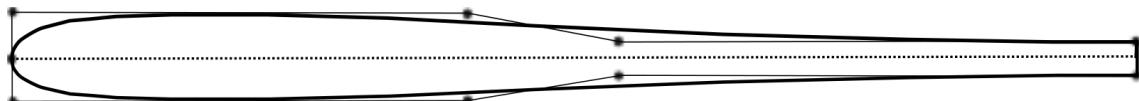


Figure 5.6: Bézier control polygons for the PS and SS.

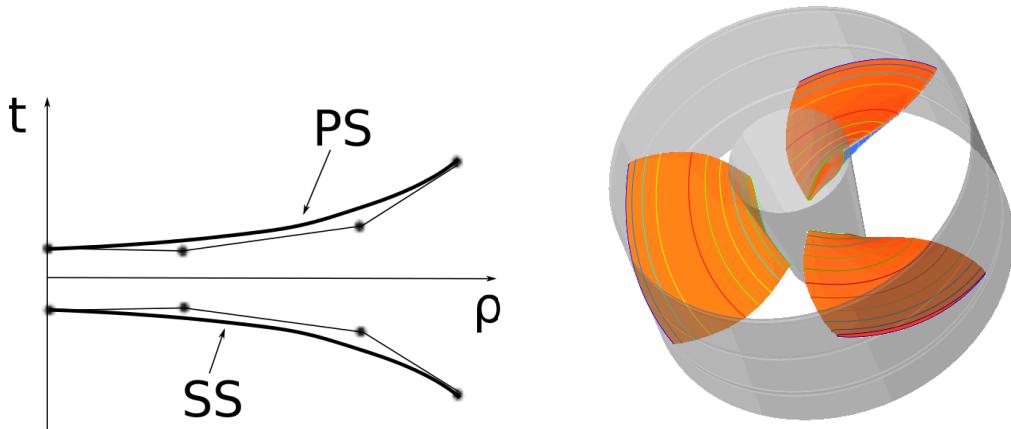


Figure 5.7: Left: $t(\rho)$ distribution. Right: 3D geometry of a Hydromatrix® turbine with three blades.

5.1.2 Space Discretization - Grid Generation

A multi-block structured grid generator, tailor-made for hydraulic turbomachines, which has been developed by Andritz-ASROE is used. This is able to handle axial, radial and mixed flow machines. Structured grids, using blocks with fixed topology can, via stretching and twisting, create grids which fit the geometries under consideration. A multi-block topology (several interconnected structure grid blocks) gives maximum freedom to the user for generating grids around quite complex geometries (fig. 5.8).

In this thesis, with respect to the topological characteristics of hydraulic turbomachines, such as the TE thickness and the extensions towards the inlet and outlet of the domain, a four-block topology is used, fig. 5.8. Block 0 is of C-type and is

5.1 Evaluation Procedure

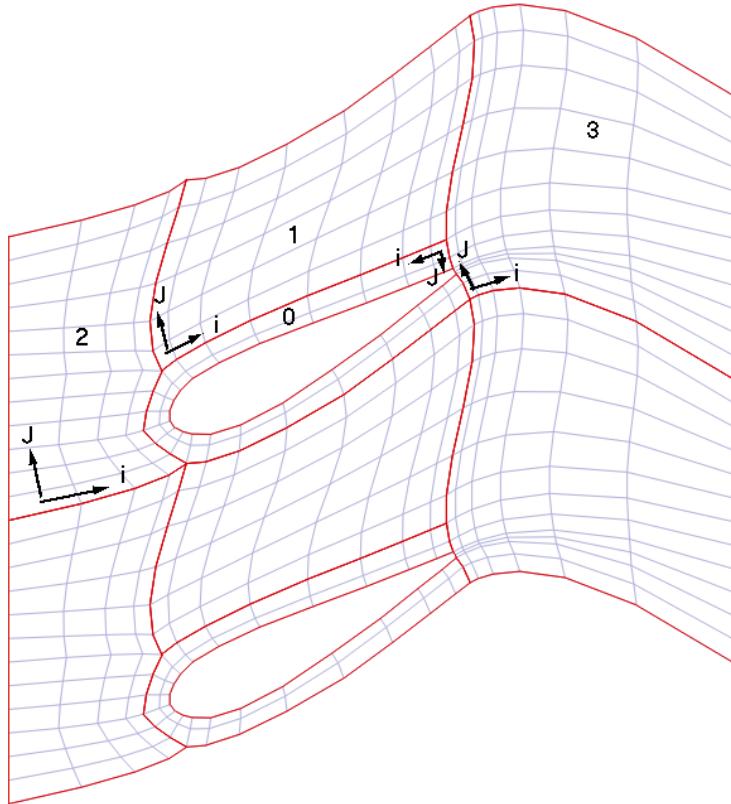


Figure 5.8: The four-block topology of a hydraulic turbine. In the sake of simplicity, this is shown for the blade-to-blade grid at mid-span. Block 0 is of C-type and is used to discretize the area in the vicinity of the blade (airfoil). Blocks 1 to 3 are of H-type and are used to fill the space from pressure side to suction side, between LE and inlet and blade TE and outlet, respectively; from (1).

used to mesh the area in the vicinity of the blade, capturing the TE thickness. Block 1 fills the space between two consecutive blades. Block 2 fills the space between the inlet and the blade LE. Finally, block 3 fills the space after blocks 0 and 1 up to the outlet.

The user may control the grid via changing the shape of the block edges, defining different numbers of nodes and/or changing the distribution of grid nodes along each boundary.

To construct the 3D grid, a user-defined number of pseudo 2D (laying on iso-span surfaces fig. 5.9) grids are computed using the grid parameterization as described above and are, then, combined to create the final 3D grid.

It is important to mention that the described grid generation technique can

5. Optimization of Hydraulic Turbomachines

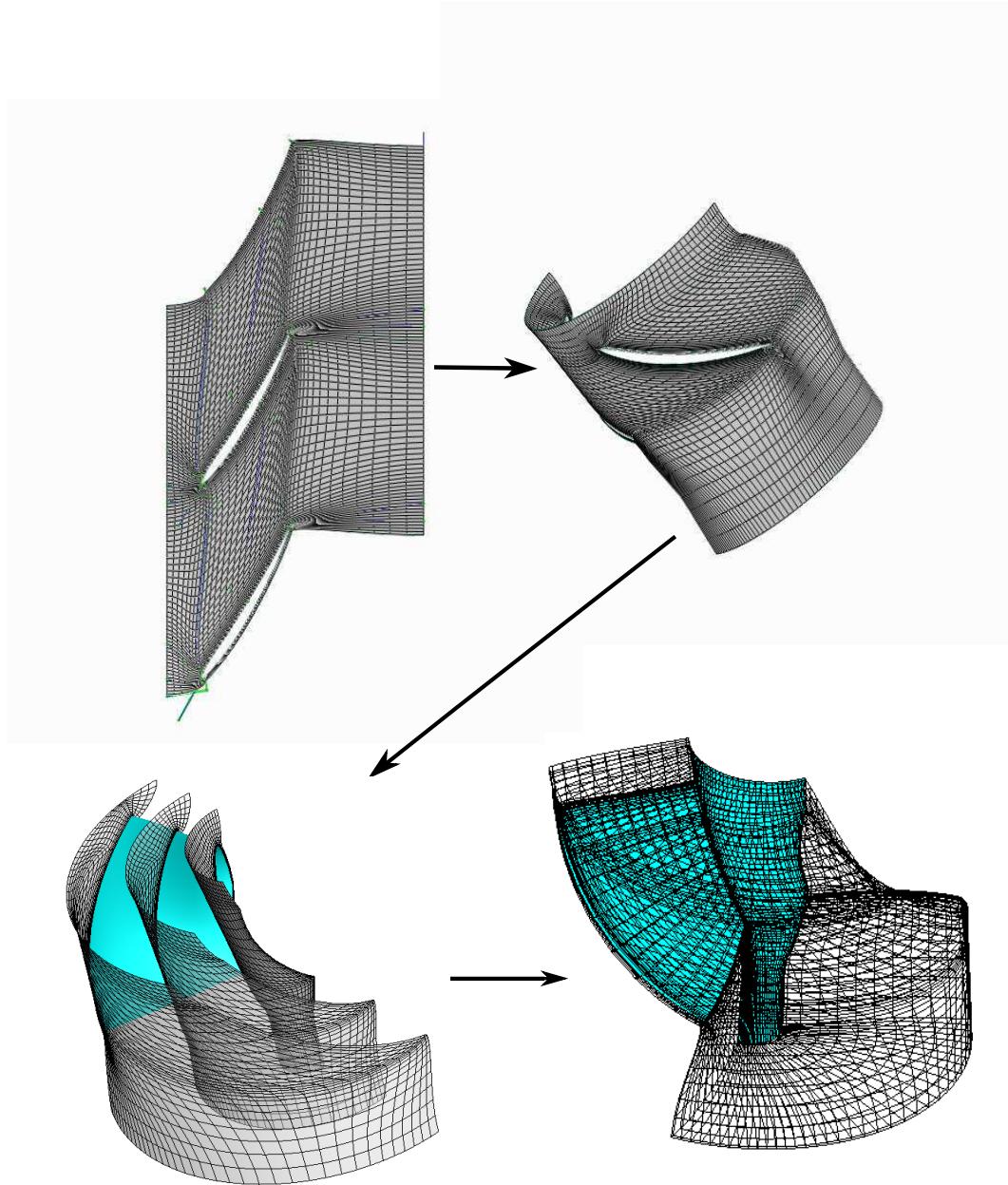


Figure 5.9: Iso-span surface grid, presented as a pseudo-2D grid. A user-defined number of the iso-span surface grids (from hub to shroud) is considered in order to generate the 3D grid. Top-Left: the pseudo-2D grid at the mid-span position. Top-Right: the same grid in the 3D space. Bottom-Left: Hub, mid-span and shroud grids. Bottom-Right: Final 3D grid.

accommodate a great variation of geometries without failing. This is of great

importance for a grid generator to be used in an optimization loop. Thus, the risk of having “good” geometries classified as “failed”, due to the failure of the grid generation process, is minimized.

In this thesis, both guide vane and runner blade channels are meshed using a grid generation tool with the previous features.

5.1.3 Flow Solver

Water flow through the passage between the hydraulic turbine blades is simulated by a CFD code developed by Andritz-ASROE. The aforementioned code is a multiblock solver for the incompressible 3D Euler equations. The Euler equations are expanded by an artificial compressibility term, (26, 178). Their discretization is based on application of the 1D Roe approximate Riemann solver, (155). The numerical solution is carried out through the ADI (Alternating Direction Implicit, (141)) method on structured grids. Convergence is accelerated via a V-cycle multigrid scheme.

In differential form, the Euler equations with the artificial compressibility term are written as

$$\frac{\partial Q}{\partial t} = \nabla \cdot F \quad (5.2)$$

where,

$$Q = \begin{pmatrix} p \\ u_1 \\ u_2 \\ u_3 \end{pmatrix}, \quad F_i = \begin{pmatrix} \beta^2 u_i \\ \beta^2 u_1 u_i + p \delta_1^i \\ \beta^2 u_2 u_i + p \delta_2^i \\ \beta^2 u_3 u_i + p \delta_3^i \end{pmatrix} \quad (5.3)$$

where β the artificial compressibility (26, 178) and δ_j^i the Kronecker symbol.

Inlet and outlet boundary conditions are applied in an explicit way, on grouped faces, via the phantom cell technique. At the inlet, the total pressure distribution and the velocity angle or the velocity profiles are imposed. At the outlet, the static pressure or the total pressure, both based on the radial equilibrium equation, are iteratively imposed.

Should the inlet velocity profile be imposed, the massflow is automatically determined and the hydraulic head results from the solution of the flow equations. On the other hand, if the total pressure profile and velocity angle are defined at the inlet in combination with the total pressure and the radial equilibrium equation at the outlet, then the hydraulic head is fixed and the massflow results from the solution of the flow equations. Such a choice plays a significant role regarding the control of the operating point, as it will be shown later on.

5. Optimization of Hydraulic Turbomachines

5.2 Hydraulic Turbine-Runner Quality Metrics

The metrics used to quantify the quality of the candidate solutions are presented below. A metric may coincide with an objective function or a part of it or may even be used as a constraint.

In this thesis, metrics are associated with:

- σ : The cavitation behaviour of the blade; cavitation is a common problem related to the operation of hydraulic machines subject to low pressure liquids, causing performance degradation and structural damages.
- M_1 : The circumferential and meridional velocity profiles at the exit position of the runner; the exit of the runner coincides with the inlet to the draft-tube and these profiles significantly affect the matching of the two components.
- M_2 : The blade loading quality, since usually the equidistribution of loading is desired along the chordwise direction of the blade at any span-wise location.
- M_3 : The pumping surface of the blade; in case of non-regulated turbines, neither the stator nor the rotor blades can be adjusted to perform optimally at every operating point. In such a case, during part-load operation, a small part of the blade surface is operating as a pump, i.e. the PS has lower pressure than the SS; the corresponding area should be minimized.

5.2.1 Cavitation Metric

In hydraulic turbines, cavitation is the phenomenon of water vaporization (formation of cavities/bubbles) as the pressure tends to become lower than the vapour pressure. When the cavitation bubbles collapse in the vicinity of a solid surface, they may cause erosion, compromising thus both the structural integrity (removed material) and the hydraulic performance (altered shape) of the turbine (24, 106, 174). In addition, the collapse of cavitation bubbles generates noise, via the large momentary pressures that are generated when the internal of the bubble is highly compressed.

The conventional way to characterize how close the minimum liquid pressure is to the vapour pressure and, therefore, quantify the danger of cavitation to occur, is by means of the so-called cavitation number σ , defined as

$$\sigma = \frac{p_\infty - p_{v,T_\infty}}{0.5\rho_L U_\infty^2} \quad (5.4)$$

where U_∞ , p_∞ and T_∞ stand for a reference velocity, pressure and temperature, respectively. ρ_L is the liquid density and p_{v,T_∞} is the saturated vapour pressure.

5.2 Hydraulic Turbine-Runner Quality Metrics

If σ is sufficiently large, p_∞ is large compared to p_{v,T_∞} or U_∞ (eq. 5.4), single-phase liquid flow will occur. However, if σ drops below the σ incipient value (σ_i), cavitation will occur. In the simplified case of a liquid which cannot withstand any tension and in which case vapour bubbles appear instantaneously when the minimum pressure (p_{min}) reaches p_v ,

$$\sigma_i = \frac{p_\infty - p_{min}}{0.5\rho_L U_\infty^2} \quad (5.5)$$

The incipient cavitation number could be ascertained from observations, measurements or simulations of single-phase flows. For $\sigma > \sigma_i$, the pressure along the entire flow field is greater than p_v thus no cavitation occurs. On the other hand, if $\sigma \leq \sigma_i$, it exists at least one point in the flow field with pressure lower than p_v where cavitation occurs.

For hydraulic turbines, the tendency of the flow to cavitate is characterized by the so-called Thoma's cavitation coefficient σ , (24)

$$\sigma = \frac{p_{t,exit} - p_v}{\rho_L g H} \quad (5.6)$$

where $p_{t,exit}$ is the total pressure at the exit of the turbine, p_v the vapour pressure for the liquid and H the hydraulic turbine head. The denominator is the total pressure difference between the turbines inlet and outlet.

Making the same assumptions as in eq. 5.5, the σ_i for the Thoma's cavitation coefficient is

$$\sigma_i = \frac{p_{t,exit} - p_{min}}{\rho_L g H} \quad (5.7)$$

Let C_p be the pressure coefficient

$$C_p = \frac{p - p_{t,exit}}{\rho_L g H} \quad (5.8)$$

The σ_i quantity for the Thoma's cavitation coefficient is

$$\sigma_i = -C_{pmin} \quad (5.9)$$

where C_{pmin} is the C_p value at the location where the minimum pressure (p_{min}) appears.

The cavitation behaviour of a hydraulic turbine can be observed through the C_p distribution, plotted over the hydraulic turbine surfaces. In fig. 5.10, the

5. Optimization of Hydraulic Turbomachines

C_p distribution of a turbine runner blade at mid-span is shown. There are , in fact, three possibilities: (a) $\sigma > -C_{pmin}$, thus $p_v < p_{min}$ and therefore cavitation doesn't occur; (b) $\sigma = -C_{pmin}$, in which case if the assumptions of eq. 5.5 and 5.7 are true, the liquid cannot withstand any tension and vapour bubbles appear instantaneously when minimum pressure (p_{min}) reaches p_v , so cavitation occurs at the point with the minimum pressure; (c) $\sigma < -C_{pmin}$, in which case $p < p_v$ over part of the blade. Therefore, this region is considered as cavitated.

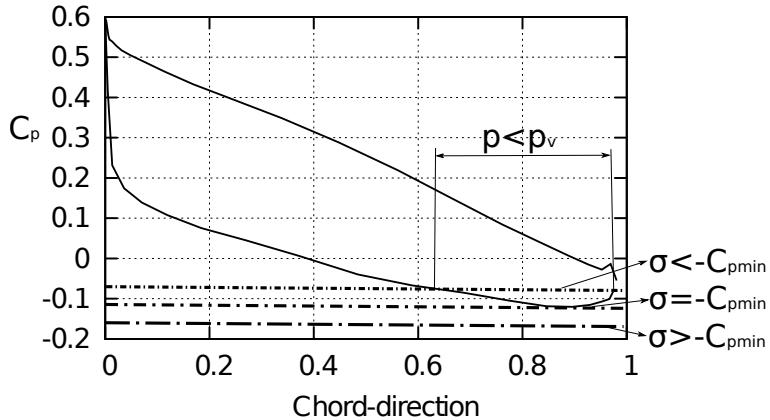


Figure 5.10: Pressure coefficient distribution at the blade mid-span.

In this thesis, in order to reduce the effects of assumptions related to eq. 5.5 and eq. 5.7, the σ -histogram method (163) is used. In the σ -histogram method, during the calculation of σ_i , the minimum pressure p_{min} is replaced with the so-called histogram pressure p_{Hist} (eq. 5.10). p_{Hist} is extracted from the blade pressure distribution, after require that a user-defined surface percentage (A) must have pressure values below p_{Hist} , (fig. 5.11)

$$\sigma_i^{Hist} = \frac{p_{tot,exit} - p_{Hist}}{\rho_L g H} \quad (5.10)$$

5.2.2 Outlet Velocity Profiles

Coupling the turbine runner with a pre-existing draft-tube is controlled via the outlet velocity profile metrics. This requires the definition of a target distribution and, thus, the corresponding metric stands for the deviation of the acquired distribution from the target one (fig. 5.12). To perform optimally, a given draft-tube must have specific mass-flow ($C_m = \frac{c_m}{\sqrt{2gH}}$) and swirl ($C_u = \frac{c_u}{\sqrt{2gH}}$) span-wise distributions, where c_m and c_u are the meridional and circumferential components of the flow velocity, respectively.

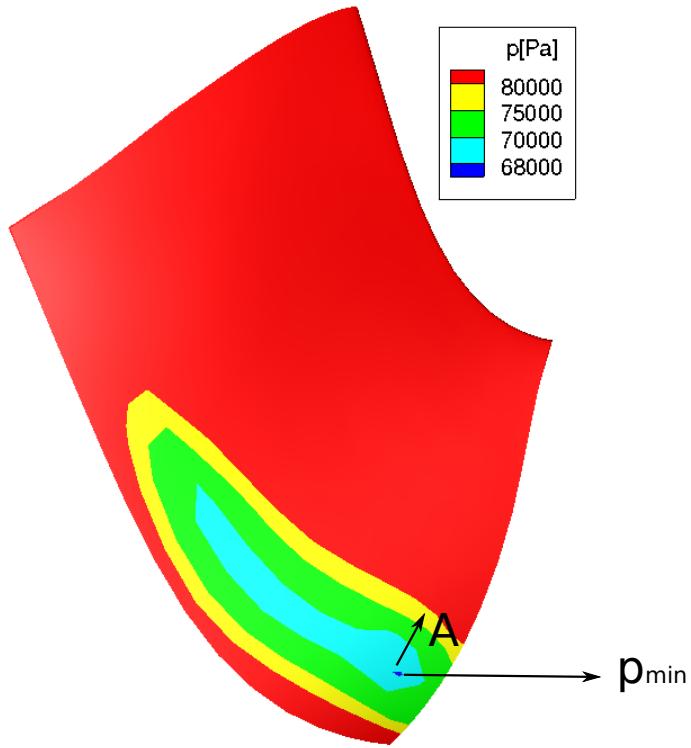


Figure 5.11: p_{Hist} computed with three different A values together with p_{min} on the SS of a Francis hydraulic turbine. Increasing A results in increased p_{Hist} and, therefore, reduced σ_i^{Hist} . The values of A used to plot this figure were artificially set quite different than the ones used in the optimization, in order to present visible iso-areas.

Therefore, the “outlet velocity quality” metric is defined as the deviation of the computed distributions from the corresponding targets.

5.2.3 Blade Loading Quality Metric

Loading quality refers to the way load (eq. 5.11) is distributed along the runner blade surface. Constant load along the blade, in the chordwise direction, is desirable (fig. 5.13) and is linked both to the hydraulic quality and the good structural behaviour. Load, at each blade position, is defined as the integral of pressure coefficient difference ($C_p^{PS} - C_p^{SS}$) over the blade surface

5. Optimization of Hydraulic Turbomachines

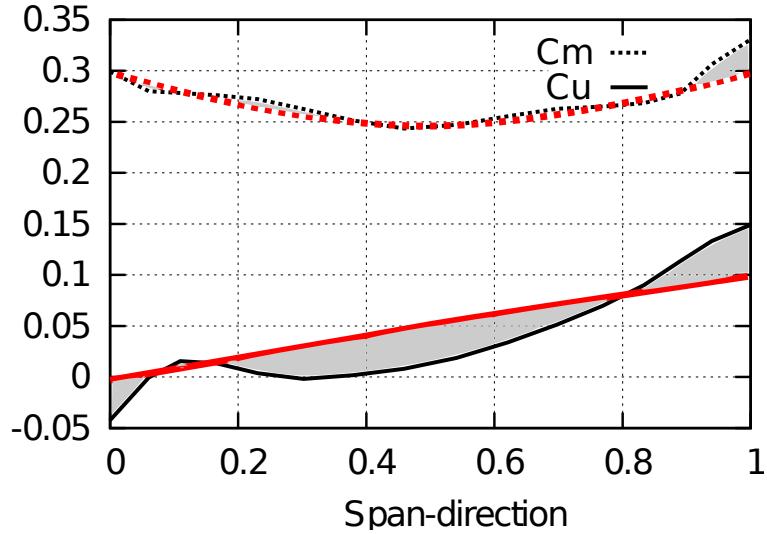


Figure 5.12: Outlet velocity distributions (thin lines) and the draft-tube specific target distributions (thick lines). C_m is shown with the dashed lines and C_u with the solid ones. The “outlet quality” metric stands for the deviation between computed distributions and targets and is marked in grey.

$$Load(x) = \int (C_p^{PS}(x) - C_p^{SS}(x))dx \quad (5.11)$$

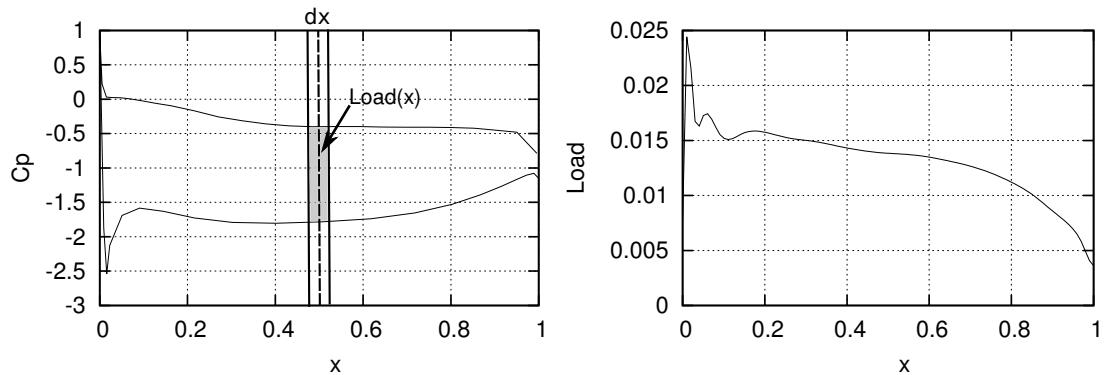


Figure 5.13: Left: Load definition. Right: Chord-wise load distributions of the profile presented on the left, as defined before.

Since constant load along the blade in the chordwise direction is desirable, optimization should aim at the minimization of the standard deviation of load along the blade surface.

5.2.4 Pumping Surface Metric

In case of non-regulated hydraulic turbines, such as the Hydromatrix® (section 5.4), an extra quality metric which is associated with their operation at part-load, must be introduced.

In non-regulated machines, neither the stator nor the rotor blades can be rotated so to adjust incoming/outgoing flow directions. During part-load operation, negative incidence angles appear (fig. 5.14), creating thus a region where the SS has higher pressures than the PS. This part of the blade is operating as a pump and, in a well performing turbine, must be minimized.

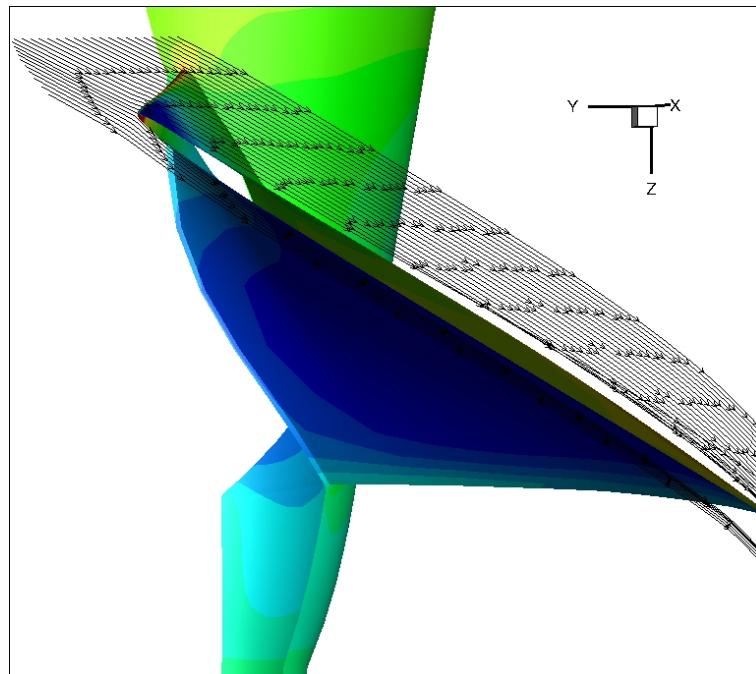


Figure 5.14: Pressure contour, along with streamlines near the shroud region, at part-load operation of a non-regulated hydraulic turbine. Water flow near the shroud region has negative incidence angle and, therefore the aforementioned pumping surface is formed. This surface needs to be minimized. The blade PS is the one on top.

In fig. 5.14, one may observe the flow streamlines at the near shroud region and how they meet the blade SS. This causes both a high-pressure region at the SS, due to the stagnation of the flow in there this meets the blade, and a low pressure region on the PS due to high flow speeds.

The partial pumping operation phenomenon can also be observed by the pressure coefficient (C_p) plots (fig. 5.15). The near-hub C_p distribution (thick continuous line) is free from pumping areas. In contrast, the near-shroud C_p distribution

5. Optimization of Hydraulic Turbomachines

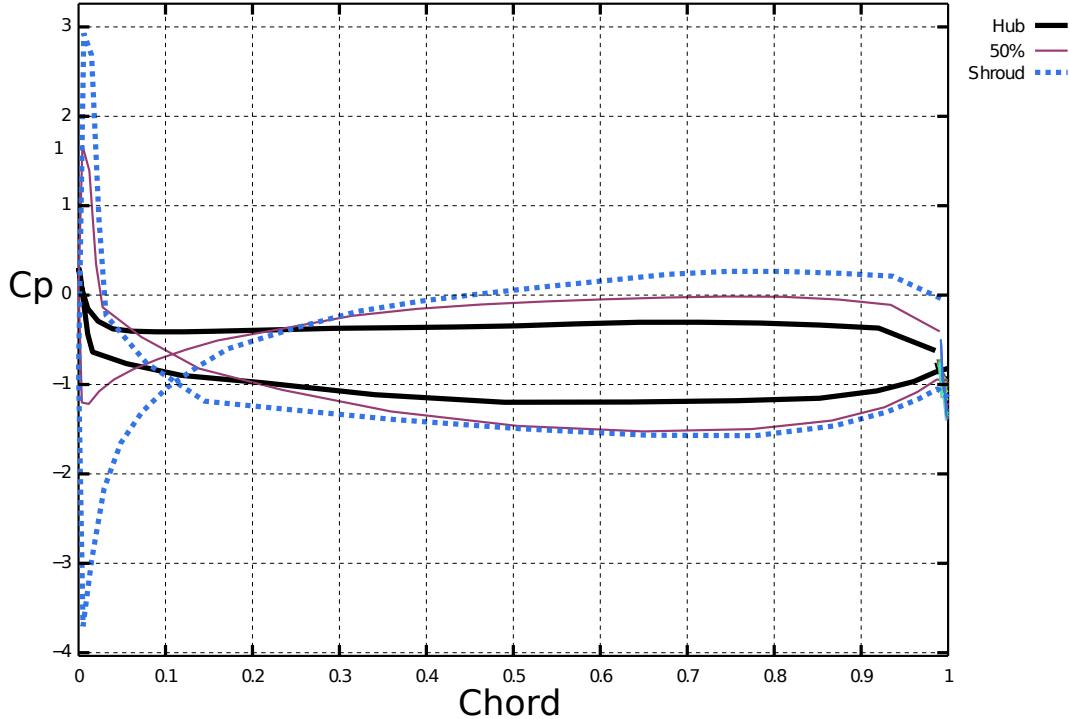


Figure 5.15: Pressure coefficient (C_p) distribution along the blade chordwise direction at part-load operation of a non-regulated hydraulic turbine at three spanwise locations: hub, mid-span and shroud. The pumping area corresponds to the first 10% of the chord at mid-span and shroud locations

(dashed line) suffers from partial pumping operation. The near-LE region of the shroud C_p distribution exhibits the so-called “crossed” distribution which denotes that the pressure at blade PS is lower than that of the SS. The thin continuous line is the mid-span C_p distribution which suffers also from pumping operation, without however being as severe as close to the shroud.

5.3 Optimization of a Francis Turbine Runner

The design-optimization of a Francis turbine runner is presented in this section. Due to the great number of design variables used to parameterize the candidate geometries, this case is perfectly suited to demonstrate the gain from the use of the KBD method.

5.3 Optimization of a Francis Turbine Runner

5.3.1 Francis Turbine

Nowadays, Francis turbines are the most frequently used water turbines since they can operate in a considerable part of the discharge-head diagram (fig. 5.16) (140). Francis is a reaction type water turbine, named after its developer James B. Francis.

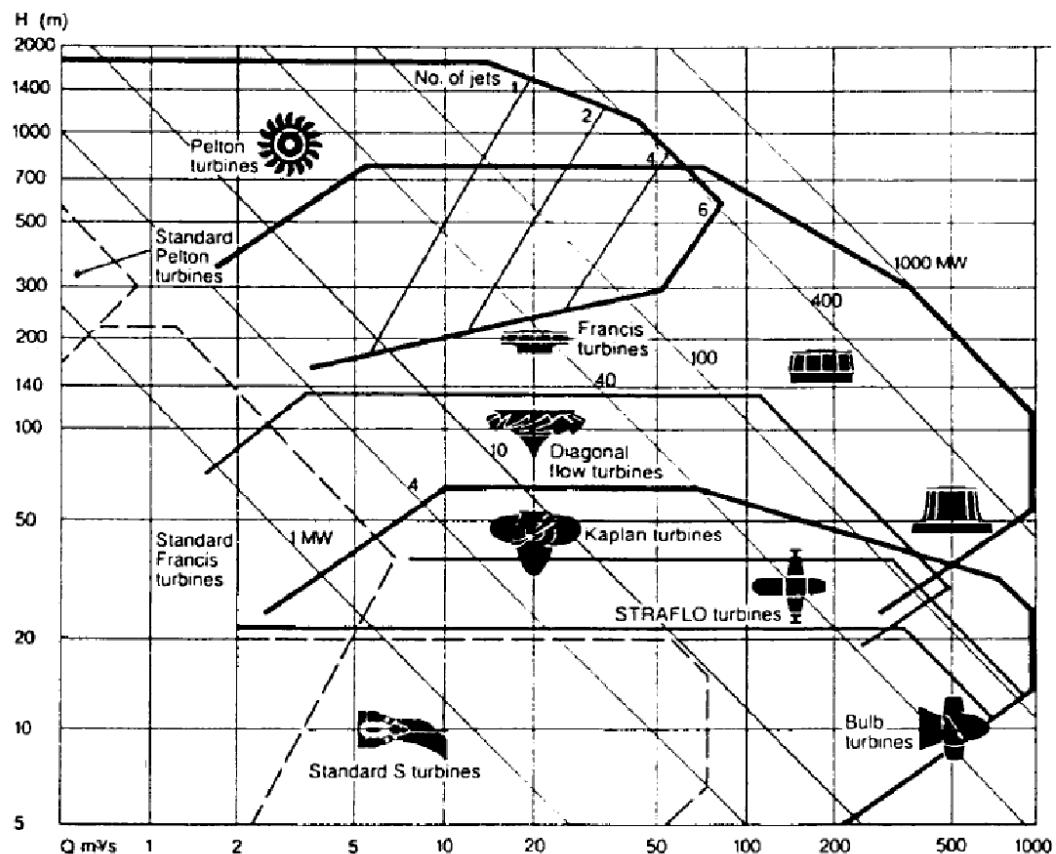


Figure 5.16: Range of application for the different turbine types, (140).

A typical Francis turbine consists of four parts (fig. 5.17). The spiral casing together with the stay-vanes are designed to provide uniform water intake along the entire circumference of the wicket-gates inlet. The wicket-gates (or guide-vanes) are adjustable to allow efficient turbine operation for a range of water flow conditions. The runner aims at harnessing the hydraulic energy. Finally, the draft-tube is used to transform the outlet kinetic energy into head.

5. Optimization of Hydraulic Turbomachines

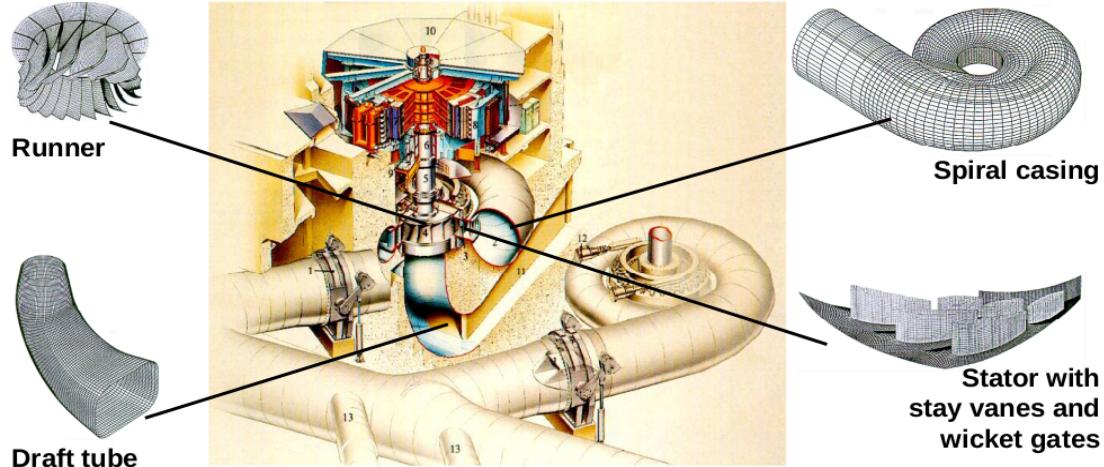


Figure 5.17: Francis turbine and its constituent parts, (2).

5.3.2 Case Presentation

From the industrial point of view, the design problem in hand is referred to as a modernization/rehabilitation project, since it mainly aims at upgrading the runner (fig. 5.18) for using it in an existing power plant. Modernization/rehabilitation projects are fairly common world-wide and, particularly in Europe where almost all large-hydro locations are already tapped and a lot of them are rather “aged”. Typical reasons for modernization/rehabilitation project are:

- (a) to increase the reliability and availability of the power plant;
- (b) to extend its life and restore its performance;
- (c) to improve its performance, including:
 - efficiency and/or power,
 - reduction of cavitation erosion,
 - enlargement of the operating range,
- (d) to improve plant safety;
- (e) to resolve environmental, social or regulatory issues;
- (f) to reduce the maintenance or operating cost;
- (g) to take into account other parameters, such as:
 - modified governmental regulations,

5.3 Optimization of a Francis Turbine Runner

- modified hydrology conditions or
- modified market conditions.

In modernization/rehabilitation projects, the designer must ensure that the newly designed runner fits perfectly in the pre-existing structure by respecting both geometrical and hydraulic-behaviour constraints. Geometrical constraints are mostly related to the inlet-outlet diameters and the hub-shroud meridional contour. On the other hand, the runner should respect given inlet flow conditions and, also, a new objective associated with the draft-tube coupling must be considered, in addition to those related to efficiency and cavitation. The reason is that, at the outlet, the runner must meet an existing draft-tube inlet which works optimally for a specific inlet velocity profile.



Figure 5.18: Optimization of a Francis Turbine: An existing runner which is quite similar to the one designed herein, (2).

5.3.2.1 Design Parameterization

The design variables and their corresponding ranges are picked from the set specified in section 5.1.1 through a user-friendly graphical interface integrated in the parameterization tool, (109, 182). In this case, the design vector consists of 336 design variables, as in table 5.1.

5. Optimization of Hydraulic Turbomachines

Number of design variables	used to parameterize the:
6	Spanwise distributions of θ_{LE}
6	Spanwise distributions of θ_{TE}
6	Spanwise distributions of β_{LE}
6	Spanwise distributions of β_{TE}
6	Spanwise distributions of ζ_{LE}
6	Spanwise distributions of ζ_{TE}
6	Spanwise thickness distributions for PS
6	Spanwise thickness distributions for SS
8	LE projection on the meridional plane
8	TE projection on the meridional plane
4	Shroud generatrix (on the meridional plane)
4	Hub generatrix (on the meridional plane)
12×11	Airfoil profiles for the PS (11 profiles)
12×11	Airfoil profiles for the SS (11 profiles)
336	Design variables, in total

Table 5.1: The design variables used to parameterize the Francis runner.

5.3.2.2 Objectives and Constraints

The objective vector comprises of two objectives which correspond to the outlet velocity profile metric (M_1) and the blade loading quality one (M_2). To define M_1 , the user-defined target profiles shown in fig. 5.19 are used.

To ensure performance stability, the runner to be designed is desirable to yield optimal performance at three operating points, the best efficiency (BE) point, the part-load (PL) and full-load (FL) ones. The three-operating-point design is handled as an optimization problem with two objectives, where each objective function is computed for each operating point separately (eq. 5.12) and, then, the “overall” value f_i is set equal to the weighted sum of the corresponding values for the three operating points, (table 5.2). Thus, if

$$f_1^i = M_1^i \quad , \quad f_2^i = M_2^i \quad (5.12)$$

where the superscript $i = 1, 2, 3$ denotes the operating point, then

$$f_1 = \sum_{i=1}^3 w_i f_1^i \quad , \quad f_2 = \sum_{i=1}^3 w_i f_2^i \quad (5.13)$$

5.3 Optimization of a Francis Turbine Runner

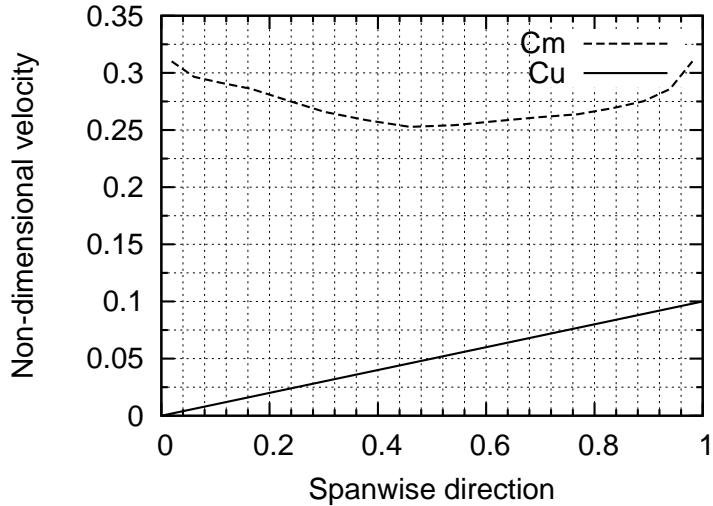


Figure 5.19: Optimization of a Francis Turbine: Target non-dimensional velocity profile distributions at the runner outlet.

where w_i the weights for each operating point (table 5.2).

Operating point	Weight (w_i)
Best Efficiency (BE), $i = 1$	1.0
Part-Load (PL), $i = 2$	0.3
Full-Load (FL), $i = 3$	0.3

Table 5.2: Optimization of a Francis Turbine: Operating point weights.

For each operating point, two constraints are imposed (table 5.3). The first is related to cavitation safety and the second aims at ensuring that the turbine operates at the desired point. The cavitation constraint is based on the σ_{Hist} method, as presented in section 5.2.1. Operating point control is carried out by imposing constraints on the maximum distance from the desired operating point. Here, given that the massflow is fixed due to the imposed inlet boundary conditions, the operating point is controlled via constraints imposed on the hydraulic height

$$\Delta H = \frac{H_{computed} - H_{desirable}}{H_{desirable}} \quad (5.14)$$

where $H_{desirable}$ the height of the desirable operating point.

5. Optimization of Hydraulic Turbomachines

Operating point	$\sigma_i^{Hist} < \sigma$	Hydraulic height
BE	$\sigma_i^{Hist} < 0.2$	$ \Delta H^i < 1.5\%$
PL	$\sigma_i^{Hist} < 0.2$	$ \Delta H^i < 5\%$
FL	$\sigma_i^{Hist} < 0.2$	$ \Delta H^i < 5\%$

Table 5.3: Optimization of a Francis Turbine: Constraints per operating point.

5.3.3 Results

As mentioned above, this case is used to demonstrate the use of the proposed KBD method in industrial environment. Therefore, two optimization procedures were tried: one by using the conventional EA since the large number of design variables makes the use of metamodels (as in standard MAEA) inefficient and a second one using the KBD-MAEA method.

5.3.3.1 Archived Designs

Three archived designs were available (fig. 5.20). These designs were developed to perform optimally at operating points close to those set for the desired blade. The criteria used to find relevant archived designs were the non-dimensional specific speed N_{ED} and specific massflow Q_{ED} ¹ defined as follows

$$N_{ED} = \frac{n}{60} \frac{D}{\sqrt{gH}}, \quad Q_{ED} = \frac{Q}{D^2 \sqrt{gH}} \quad (5.15)$$

The archived designs to be used as basis vectors for this case were re-evaluated at the desirable operating points and their quality metrics are summarized in table 5.4.

A closer look at B1 reveals the findings tabulated in table 5.4. For instance, from fig. 5.21, where C_p iso-areas are plotted over the runner surfaces, areas which are prone to cavitation can be identified. From this figure, one may observe the lowest value of $C_p = -0.216$ which advocates the presence of cavitation, since $\sigma_i \approx -C_p = 0.216$. Recall that $\sigma = 0.2$ and that σ_i must be greater than σ for cavitation to appear. The σ_i^{Hist} value which is equal to 0.21, confirms this observation. Recall that σ_i^{Hist} is a better estimation of σ_i than the $-C_p$ one. So, the archived design B1 is suffering from cavitation even at the BE point. Fig. 5.21 shows that regions with low C_p which are prone to cavitation exist over the SS, near the TE, between mid-span and shroud. The archived design B1 suffers

¹ N_{ED} and Q_{ED} are dimensionless quantities based on energy (E=1) and diameter (D=1) used to characterize turbine operation.

5.3 Optimization of a Francis Turbine Runner

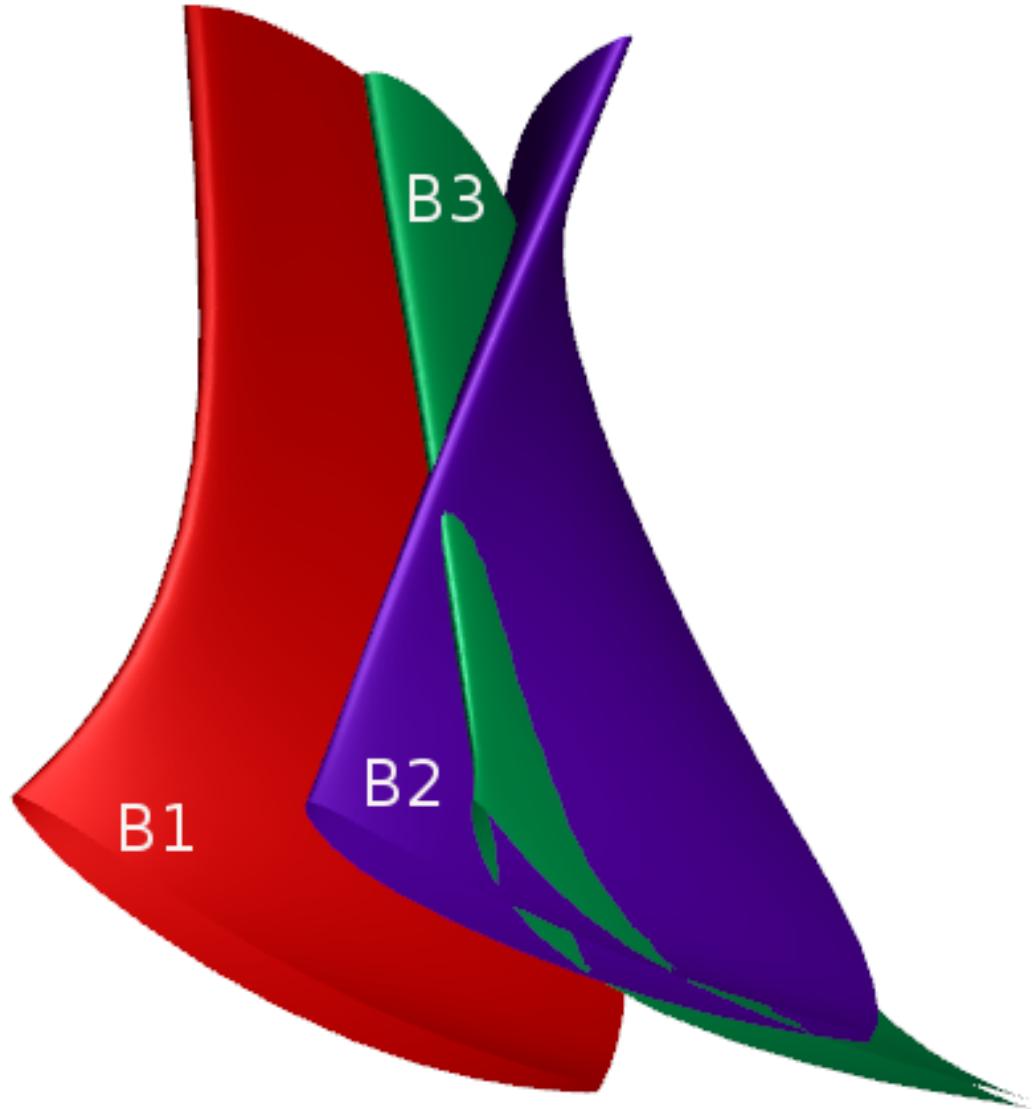


Figure 5.20: Optimization of a Francis Turbine: The three archived geometries used as basis vectors.

from cavitation at the PL point too and is just marginally safe at the FL point (table 5.4).

Regarding the same design B1, fig. 5.22 shows the noticeable deviation from the desirable outlet C_m and C_u velocity profiles for operating at BE point. Dashed

5. Optimization of Hydraulic Turbomachines

Archived Design	OP	M_1	M_2	σ_i^{Hist}	$ \Delta H $
B1	BE	0.472	0.538	$0.21 > 0.2 *$	$5\% > 1.5\% *$
	PL	0.552	0.824	$0.23 > 0.2 *$	$3.6\% < 5\%$
	FL	0.408	0.594	$0.2 = 0.2$	$5.7\% > 5\% *$
B2	BE	0.563	0.922	$0.22 > 0.2 *$	$3.8\% > 1.5\% *$
	PL	0.534	0.888	$0.24 > 0.2 *$	$2.6\% < 5\%$
	FL	0.522	1.178	$0.22 > 0.2 *$	$4.8\% < 5\%$
B3	BE	0.402	0.663	$0.19 < 0.2$	$6.8\% > 1.5\% *$
	PL	0.515	0.808	$0.21 > 0.2 *$	$6.9\% > 5\% *$
	FL	0.257	0.896	$0.18 < 0.2$	$7.1\% > 5\% *$

Table 5.4: Optimization of a Francis Turbine: Quality metrics (objectives and constraints) for the 3 archived designs (B1, B2 and B3) for all operating points (BE, PL and FL). An asterisk (*) next to an inequality means that the corresponding constraint is violated.

lines represent C_m and continuous lines C_u . For the purpose of comparison, in the same plot, the target distributions are plotted with thicker lines.

The loading quality of B1 can be demonstrated through the chord-wise C_p distribution plot, as defined in eq. 5.8, along the hub, mid-span and shroud (fig. 5.23). Though, at mid-span and hub, relatively good loading behaviour is observed, the blade part which is close to the shroud suffers from important load differences.

5.3 Optimization of a Francis Turbine Runner

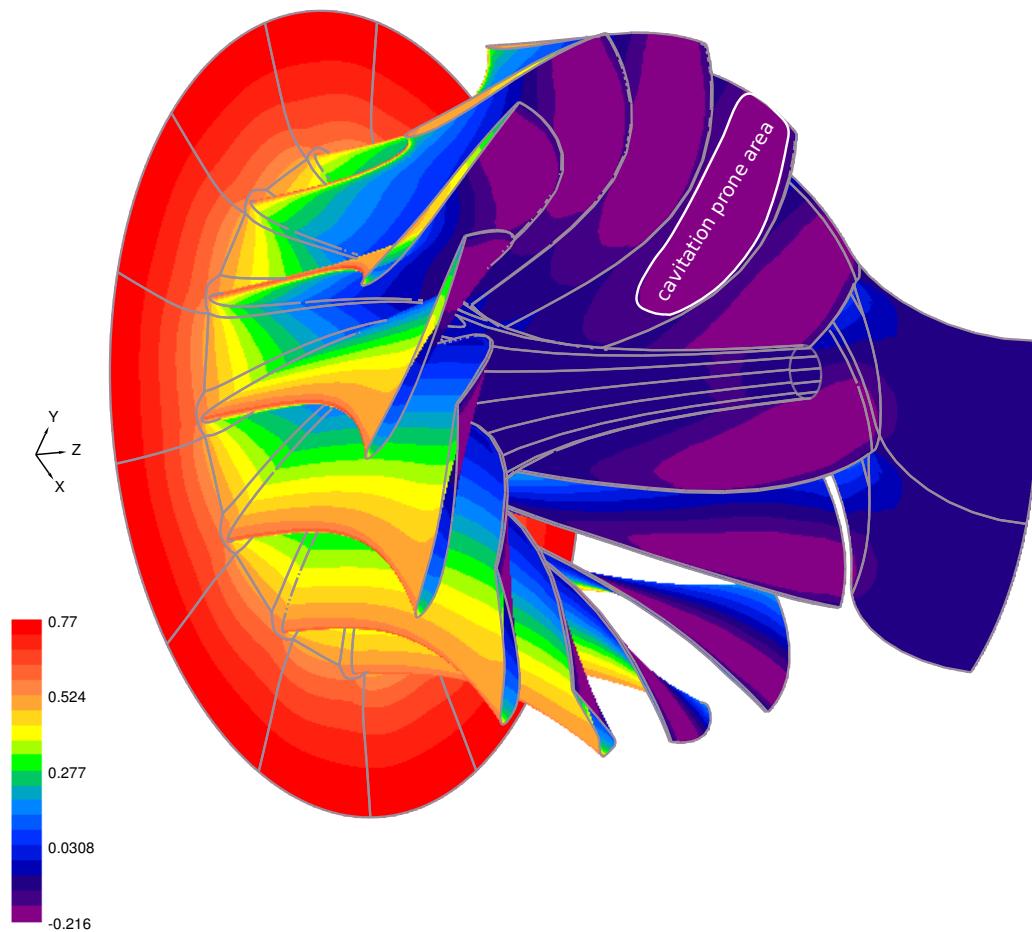


Figure 5.21: Optimization of a Francis Turbine: C_p iso-areas for the archived geometry B1, at the BE operating point. The area which is prone to cavitation (region with low C_p) is clearly marked at SS, near the TE, between mid-span and shroud.

5. Optimization of Hydraulic Turbomachines

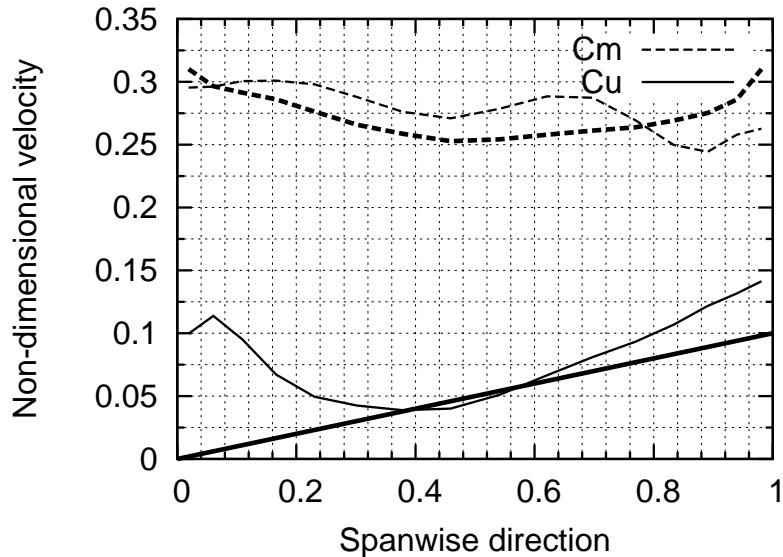


Figure 5.22: Optimization of a Francis Turbine: C_m and C_u profiles for the archived design B1, operating at the BE point. Target distributions are plotted with thicker lines.

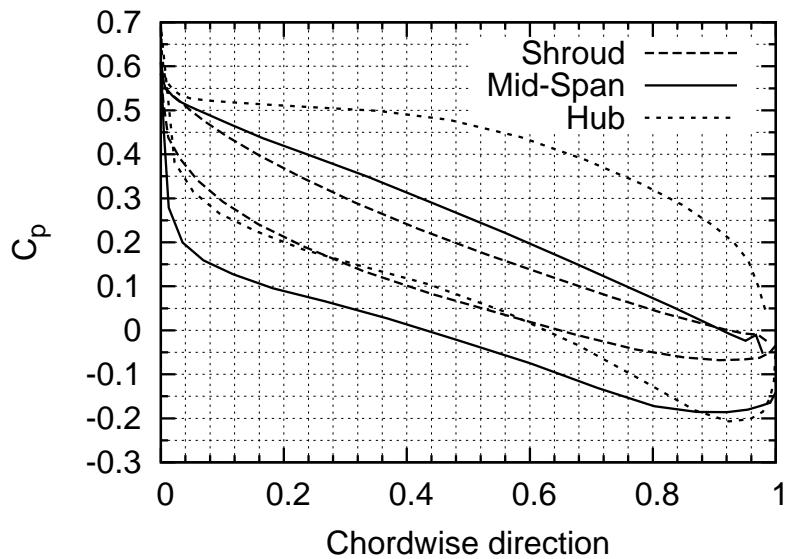


Figure 5.23: Optimization of a Francis Turbine: C_p profiles hub, mid-span and shroud for the archived design B1, for the BE operating point.

5.3 Optimization of a Francis Turbine Runner

Regarding B2, the existing blade suffers from cavitation at all three operating points, see table 5.4. Fig. 5.24 shows the C_p iso-areas over the runner blade surface of B2, operating at the BE point. The lower limit of C_p , ($C_p = -0.234$) denotes that cavitation might appear and this is confirmed by the computed σ_i^{Hist} values (table 5.4). Fig. 5.24 shows that cavitation occurs over the suction side near the trailing edge at the near hub region.

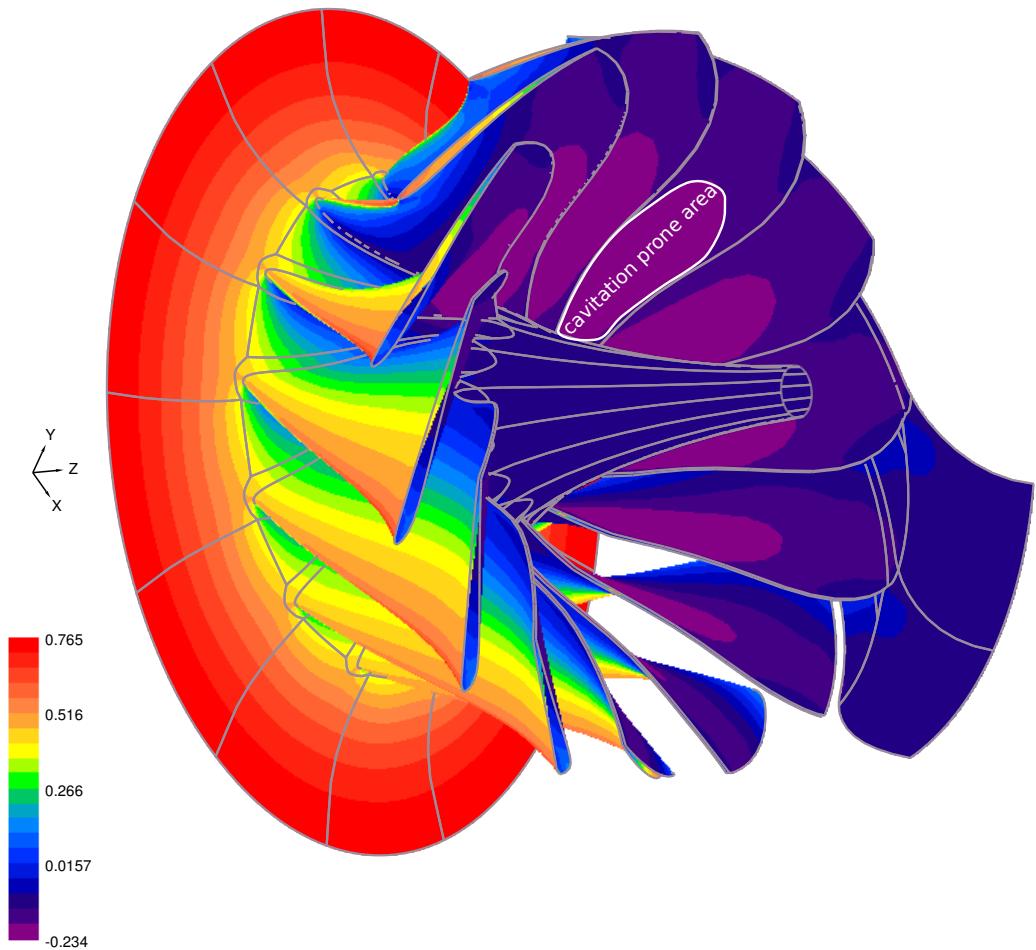


Figure 5.24: Optimization of a Francis Turbine: C_p iso-areas for the archived design B2 operating at the BE point. Cavitation occurs on SS, near the TE close to the hub.

The outlet velocity quality is plotted in fig. 5.25. C_m is far away from the desirable distribution and C_u , close to the hub, has locally high values.

5. Optimization of Hydraulic Turbomachines

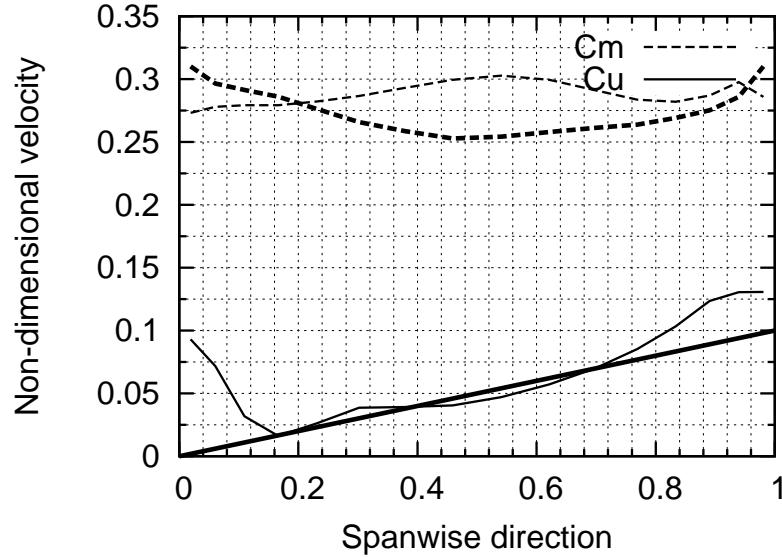


Figure 5.25: Optimization of a Francis Turbine: C_m and C_u profiles for the archived design B2 at the BE operating point. Target distributions are plotted with thicker lines.

From the C_p profiles computed for the archived design B2 operating at the BE point (fig. 5.26), one may observe the non-satisfactory loading quality; the same information can be obtained from table 5.4. In the three spanwise positions, there are important load differences along the chord direction.

5.3 Optimization of a Francis Turbine Runner

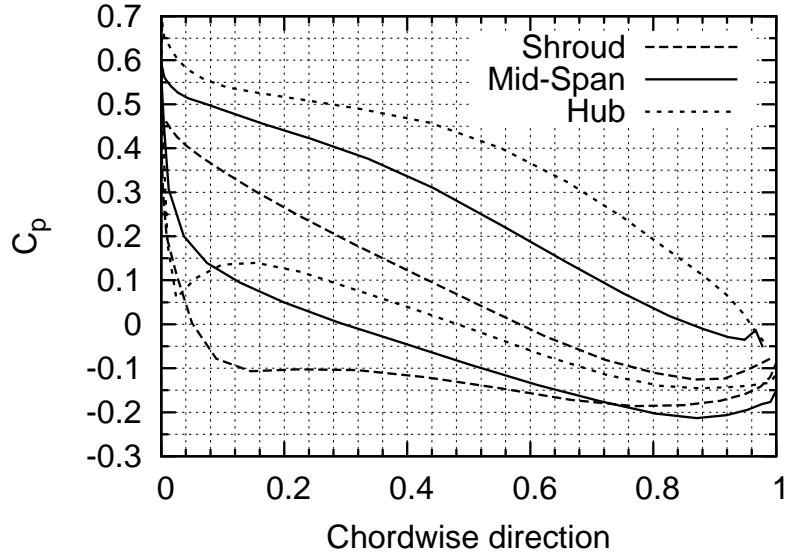


Figure 5.26: Optimization of a Francis Turbine: C_p profiles at hub, mid-span and shroud for archived design B2 at the BE point.

Archived design B3 is clean from cavitation at all but the PL operating point, see table 5.4. Fig. 5.27 shows the C_p iso-areas for B3 operating at the PL point. The lowest C_p value observed, namely $C_p = -0.215$, is a clear indication of the presence of cavitation, as also confirmed by the computed σ_i^{Hist} value (table 5.4). Cavitation is observed at the suction side near the TE.

Regarding the outlet velocity profile quality, fig. 5.28, both C_m and C_u deviate considerably from the desirable distributions.

Loading quality is demonstrated in fig. 5.29. The computed C_p profile at the shroud shows important loading differences.

Since none of the three archived designs shows “good” outlet velocity profiles and loading and, also, doesn’t respect the imposed constraints, a “Revise” step is needed. In this thesis, this step is based on EAs, with or without metamodels. So, practically, two optimizations have been carried out. The first was based on a conventional EA ($\mu = 30, \lambda = 120$) using 336 design variables, according to the aforementioned parameterization. The second employed the proposed KBD method, i.e. the so-called KBD-MAEA ($\mu = 20, \lambda = 60, \lambda_e = 4 \text{ to } 8$, with the IPE phase starting after recording the first 150 non-failed individuals in the DB). The KBD-MAEA run handled 19 optimization variables. For both optimizations, the three archived designs were injected in the population of the first generation.

Regarding KBD, the 19 optimization variables result from grouping the design variables (eq. 3.6) into 6 groups (table 5.5) plus the extrapolation variable Ψ . This set-up requires 6 weights to control the influence of each one of the archived

5. Optimization of Hydraulic Turbomachines

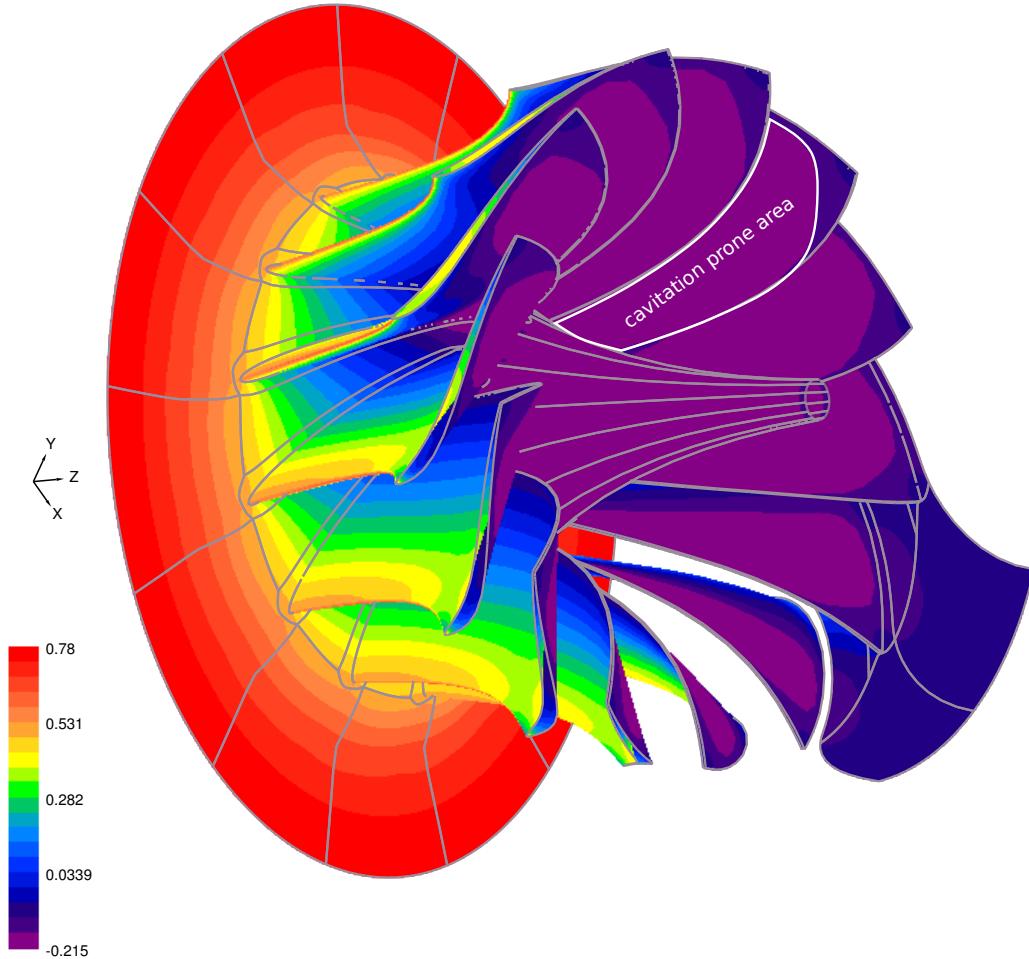


Figure 5.27: Optimization of a Francis Turbine: C_p iso-areas for archived design B3 at PL operating point. Cavitation is observed at SS, near the TE.

designs to any candidate solution, giving rise to 18 weights, in total.

The comparison of the performance of EA and KBD-MAEA is made using the hypervolume indicator (185). The hypervolume indicator assumes that the quality of a Pareto front can be expressed by a scalar value which stands for the area or volume or hypervolume of the dominated, by the Pareto front, part of the objective space up to a user-defined nadir point. With the same nadir point, the higher the hypervolume indicator, the higher the quality of the Pareto front.

The comparison of the evolution of the hypervolume indicator in terms of the number of exact evaluations carried out by the EA and the KBD-MAEA is

5.3 Optimization of a Francis Turbine Runner

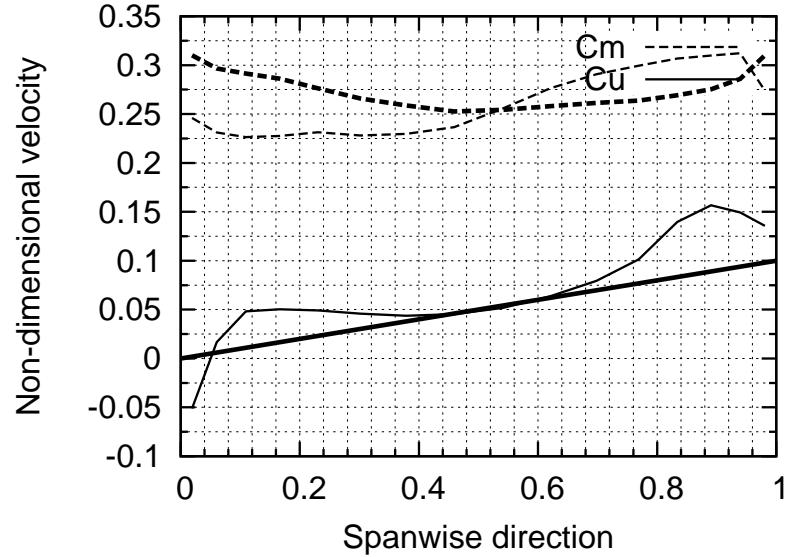


Figure 5.28: Optimization of a Francis Turbine: C_m and C_u profiles for the archived design B3, at the BE point. For the purpose of comparison, in the same plot, target distributions are plotted with thicker lines.

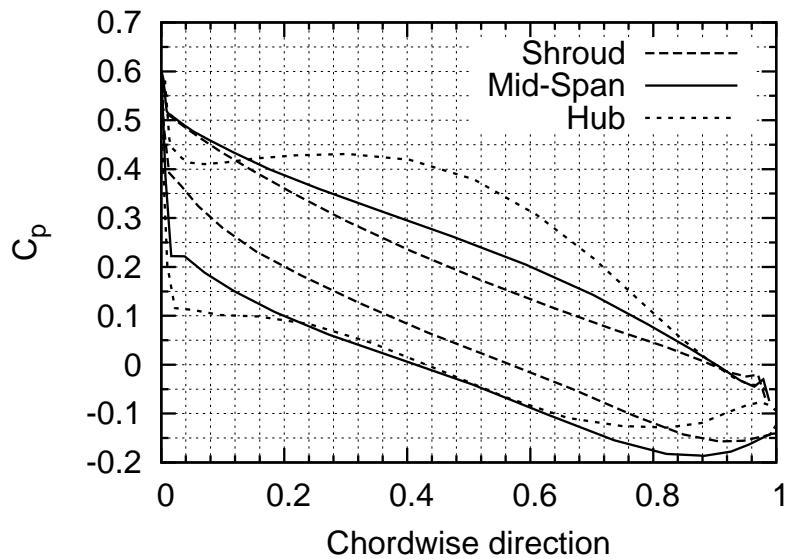


Figure 5.29: Optimization of a Francis Turbine: C_p profiles hub, mid-span and shroud, for the archived design B3, at the BE point.

shown in fig. 5.30. The horizontal axis expresses the CPU cost, by just making the realistic assumption that all evaluations have the same cost.

5. Optimization of Hydraulic Turbomachines

Group	Design variables determining the:
1	Spanwise distributions of θ_{LE}
1	Spanwise distributions of θ_{TE}
2	Spanwise distributions of β_{LE}
2	Spanwise distributions of β_{TE}
3	Spanwise distributions of ζ_{LE}
3	Spanwise distributions of ζ_{TE}
4	Spanwise thickness distributions for PS
4	Spanwise thickness distributions for SS
5	LE meridional position
5	TE meridional position
5	Shroud meridional generatrices
5	Hub meridional generatrices
6	Airfoil profiles for the PS (11 profiles)
6	Airfoil profiles for the SS (11 profiles)
6	Groups, in total

Table 5.5: Optimization of a Francis Turbine: The 6 Groups defining the $6 \times 3 = 18$ optimization variables out of the 19 used in the KBD method.

The KBD-MAEA significantly outperforms the conventional EA. It starts with significantly better individuals in the first generation which demonstrates that, even though the archived designs are far from the optimal design, the grouping of the design variables and the use of normal distribution to set the importance regions in the design space are very close to the “physics” of the problem in hand.

By examining fronts of non-dominated solutions (fig. 5.31), as computed by the two methods at the same CPU cost of 1500 exact evaluations, it can easily be concluded that designs of higher quality are obtained by the KBD-MAEA run. One of the non-dominated solutions computed by the KBD-MAEA, i.e. the one marked with A, was chosen for further analysis at the three operating points.

The quality metrics of design A are summarized in table 5.6. To compare design A with the archived ones, its metrics should be compared with the values presented in table 5.4. This comparison reveals that, design A is significantly better than the three archived designs at all operating points and for both objectives. Design A is, in fact, a high quality blade.

Regarding cavitation, the behaviour of the selected design A can be seen by examining figs. 5.33, 5.34 and 5.35, for the BE, PL and FL points, respectively.

At the BE point, the lowest C_p value is -0.179 , fig. 5.33, which is a clear indication of the absence of cavitation. At the PL point, the lowest C_p value

5.3 Optimization of a Francis Turbine Runner

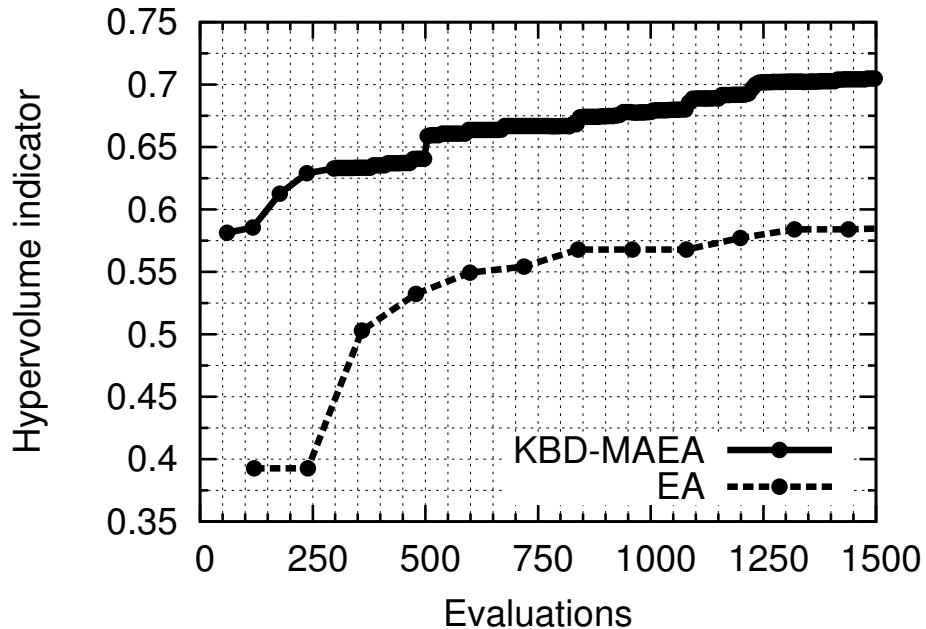


Figure 5.30: Optimization of a Francis Turbine: hypervolume comparison between EA and the proposed KBD-MAEA method.

Geometry	OP	M_1	M_2	σ_i^{Hist}	$ \delta H $
A	BE	0.001	0.302	$0.18 < 0.2$	$1.1\% < 1.5\%$
	PL	0.086	0.409	$0.19 < 0.2$	$1.5\% < 5\%$
	FL	0.092	0.504	$0.18 < 0.2$	$4.1\% < 5\%$

Table 5.6: Optimization of a Francis Turbine: Quality metrics (objectives and constraints) regarding the chosen design A for all operating points (BE, PL & FL). The design is safe regarding cavitation at all operating points and operates within the desirable range.

is -0.192 and, therefore, $\sigma_i \approx 0.192$ which is quite close to the 0.2 limit but still safe. Typically, off-design points, such as the PL and the FL ones, operate marginally closer to the cavitation limit than the BE one. Design A, operating at PL yields the lowest values of C_p at the SS of the blade near the TE, fig. 5.34. At the FL point, the lowest C_p value is equal to -0.182 (fig. 5.35). Fig. 5.36 demonstrates that the blade areas which are closer to cavitation are located close to the TE and LE, on the SS.

Furthermore, the outlet flow quality of design A at the BE point is plotted in fig. 5.37. Design A clearly outperforms all three archived designs regarding the outlet flow quality metric. A quite good agreement with the desirable outlet C_m

5. Optimization of Hydraulic Turbomachines

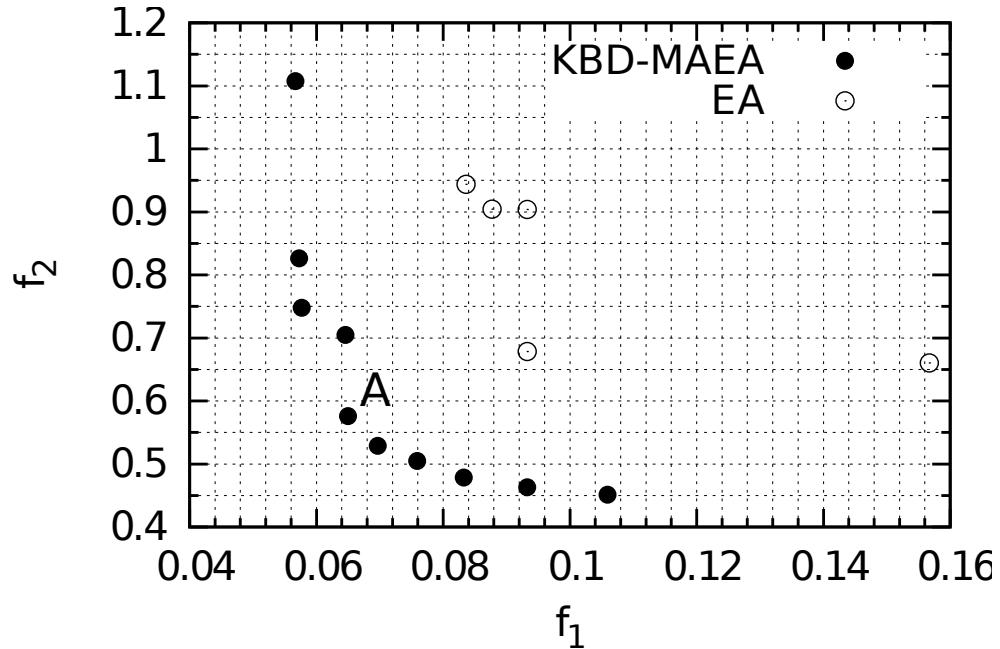


Figure 5.31: Optimization of a Francis Turbine: Fronts of non-dominated solutions computed at the cost of 1500 exact evaluations by the EA and KBD-MAEA method. Optimal solution A was selected for further analysis (see figs. 5.32 to 5.38).

and C_u distributions is obtained.

For the same design, its loading quality at hub, mid-span and shroud is presented in fig. 5.38.

5.3 Optimization of a Francis Turbine Runner

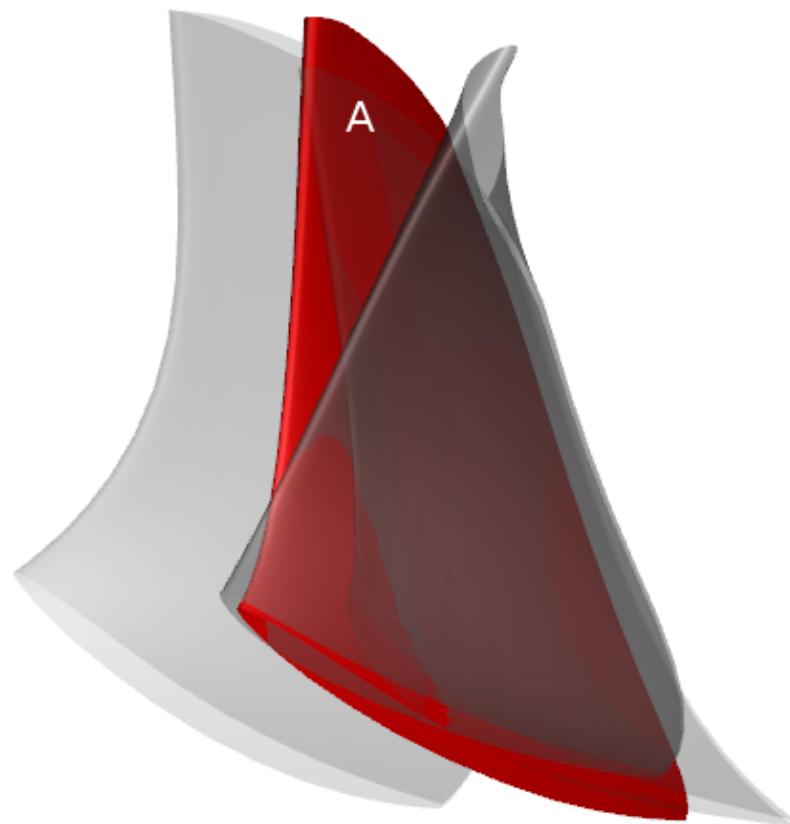


Figure 5.32: Optimization of a Francis Turbine: Design A and the three archived designs used. The high diversity among the four designs can be observed. The ability of the KBD method to produce designs, highly diverse in comparison with the archived ones is noticeable.

5. Optimization of Hydraulic Turbomachines

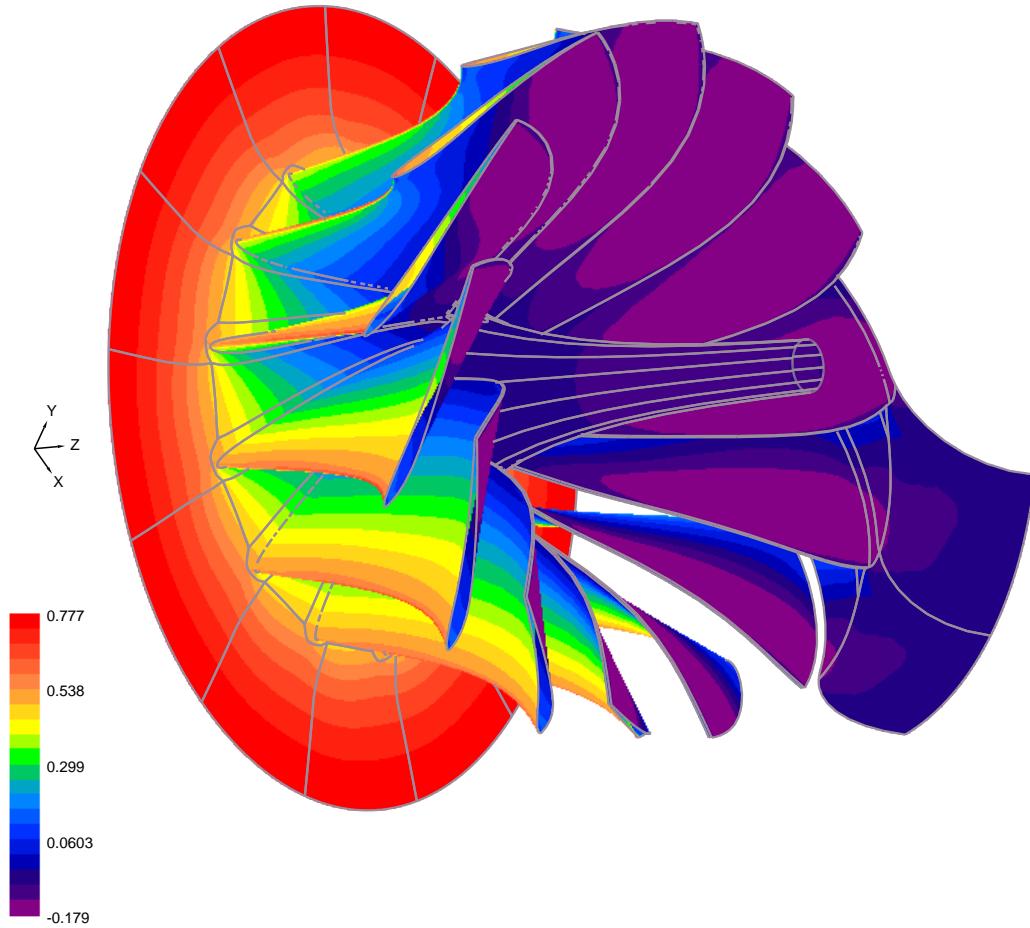


Figure 5.33: Optimization of a Francis Turbine: C_p contour for design A at the BE point. The lowest C_p value is equal to -0.179 corresponding to a σ_i value of 0.179 , well below the imposed cavitation limit (0.2).

5.3 Optimization of a Francis Turbine Runner

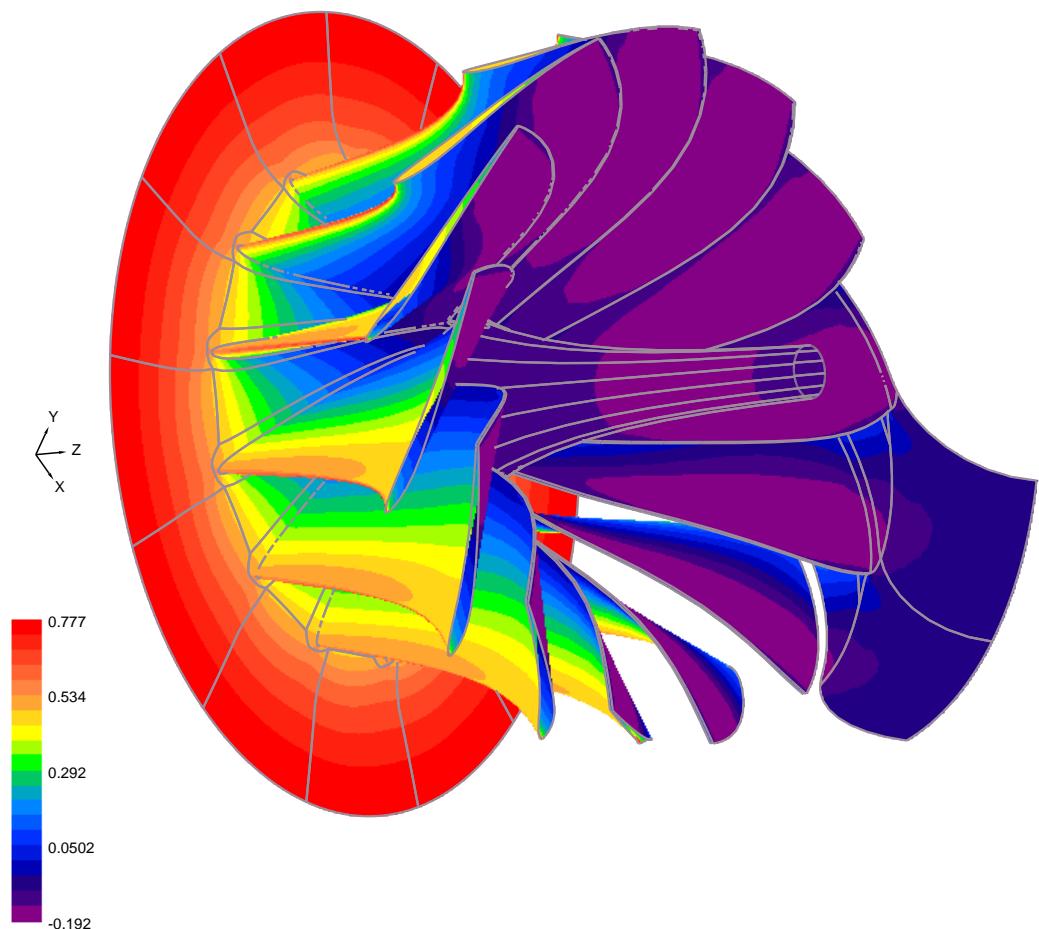


Figure 5.34: Optimization of a Francis Turbine: C_p contour for design A at the PL point. The lowest C_p value is equal to -0.192 which compared to the BE point operation, fig. 5.33, is quite close to the cavitation threshold but still on the safe side.

5. Optimization of Hydraulic Turbomachines

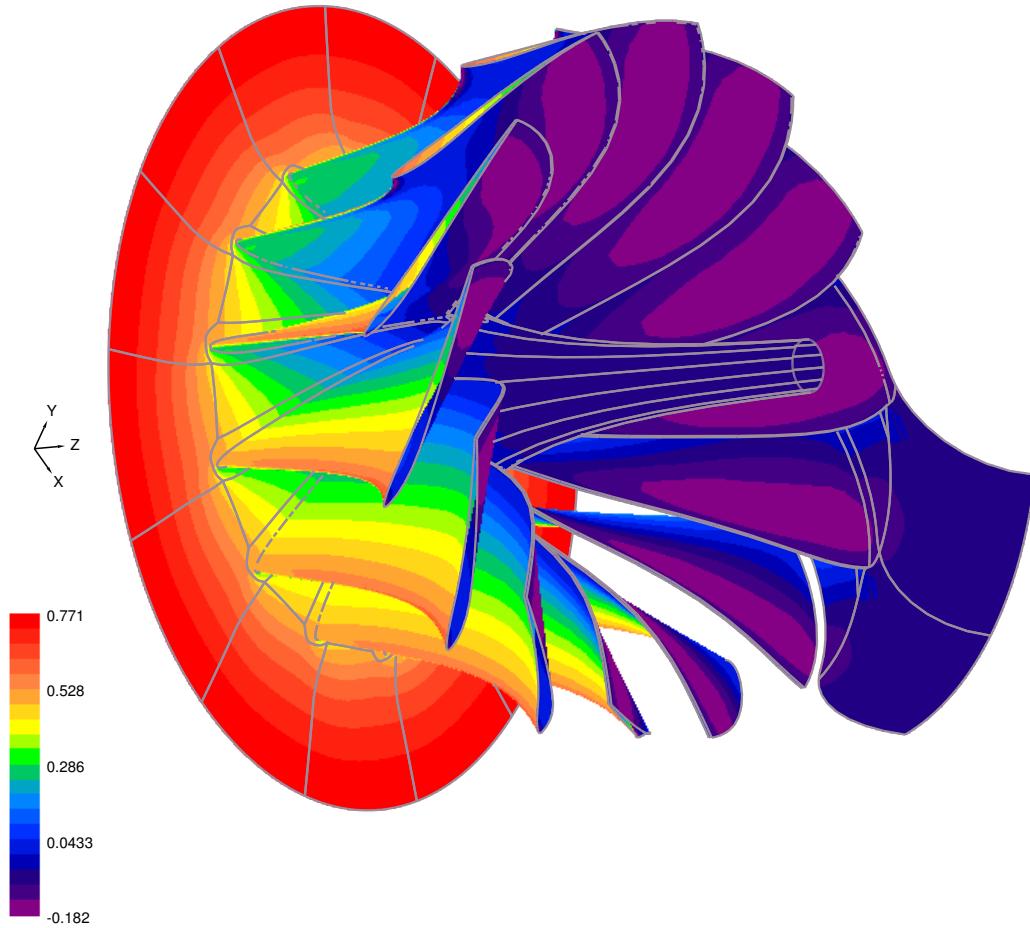


Figure 5.35: Optimization of a Francis Turbine: C_p contour for design A at the FL point. The lowest C_p value is -0.182 , which is lower than that for the operation at the BE point. However, this is higher than that of the PL point.

5.3 Optimization of a Francis Turbine Runner

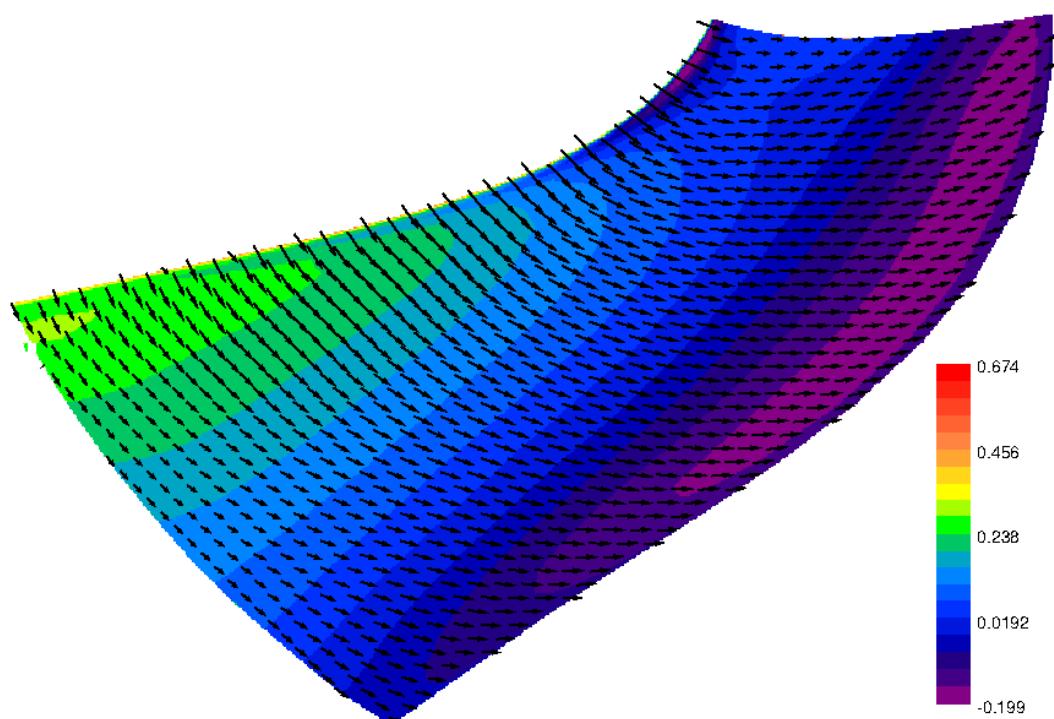


Figure 5.36: Optimization of a Francis Turbine: C_p contour for design A at the FL operating point over SS. Over the SS of the blade operating at FL, there is an additional low C_p zone near the LE, in addition to that existing close to the TE.

5. Optimization of Hydraulic Turbomachines

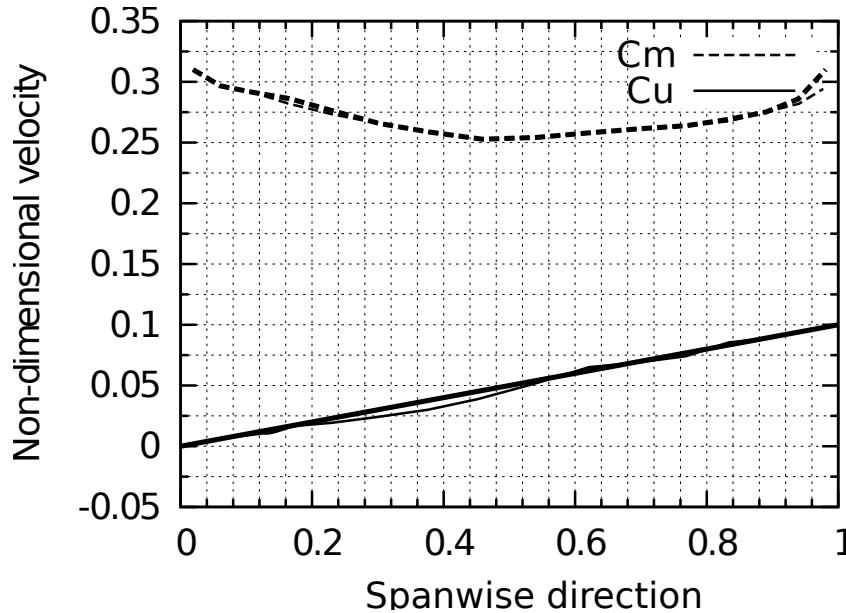


Figure 5.37: Optimization of a Francis Turbine: Outlet C_m and C_u profiles for design A at the BE point.

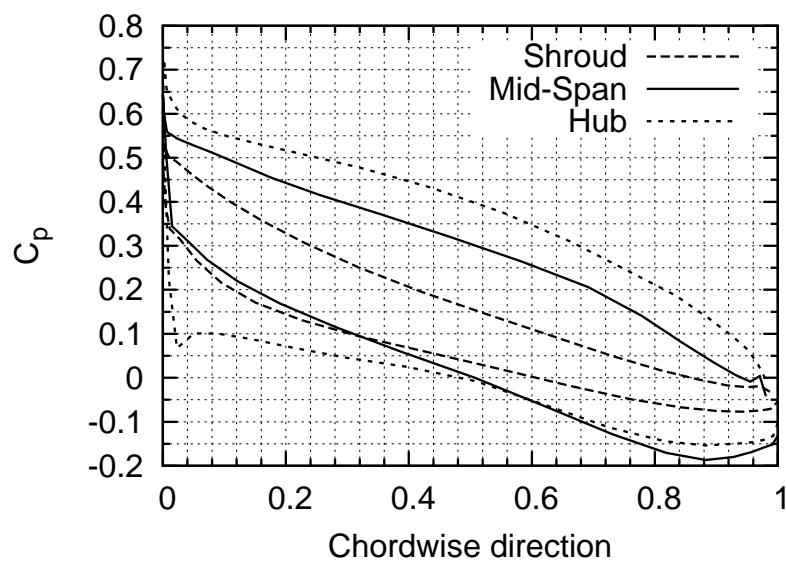


Figure 5.38: Optimization of a Francis Turbine: C_p profiles at hub, mid-span and shroud, for design A at the BE point.

5.4 Optimization of a Hydromatrix® Turbine

In this section, the design-optimization of a complete Hydromatrix® turbine is presented. This case is used to demonstrate the expected gain from the use of the proposed MAEA(PCA). Many industrial optimization problems, such as the design of a Hydromatrix® turbine, are ill-posed (in the way this term is used in this thesis) which, as shown in Chapter 4, causes drop in EAs efficiency.

5.4.1 The Hydromatrix® Turbine

Hydromatrix® is an axial reaction type water turbine (fig. 5.39) developed by Andritz Hydro as an innovative solution for use in low-head hydropower sites (3, 4). The concept of Hydromatrix® relies on the use of a number of small non-regulated (fixed blade) turbines, in place of a conventional large regulated one (fig. 5.40). The proposed way to regulate a Hydromatrix® system is via closing/opening a number of turbines so to keep the massflow per open turbine as close as possible to that of its BE operating point. In order to avoid flood, if the flow exceeds the turbines capacity, a number of turbines can be raised or removed from their operating positions like a gate (fig. 5.40).

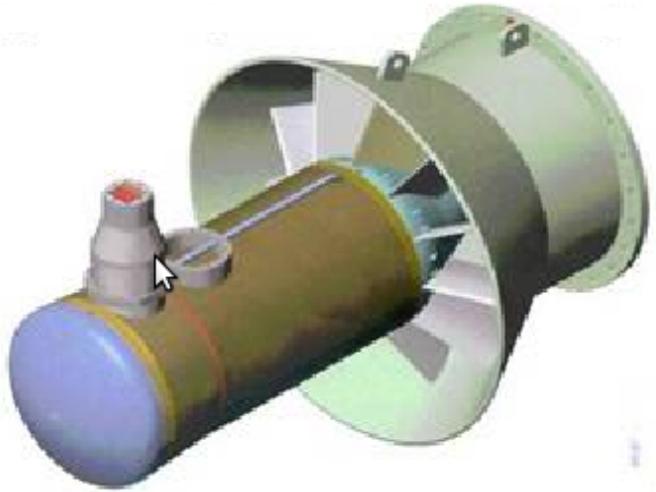


Figure 5.39: Optimization of a Hydromatrix® Turbine: Hydromatrix turbine module, (3, 4).

The Hydromatrix® ability to use existing weir structures reduces the additional civil works to minimum and enables power plant operators to install hydroelectric power plants at extremely competitive costs with minimal environmental impact. The number of matrix turbines and their arrangement in rows

5. Optimization of Hydraulic Turbomachines

depends on the existing civil structure and its position relative to the head- and tail-water elevations. The use of a number of small turbine generator units (also known as “modules”) results in simple and robust turbine and generator designs. The standardized modular factory assembled grid or “matrix” (fig. 5.40) results in short project schedules and high availability. A Hydromatrix® turbine consists of three different parts: the distributor cone containing the stay-wanes, the runner and the draft-tube (fig. 5.39).

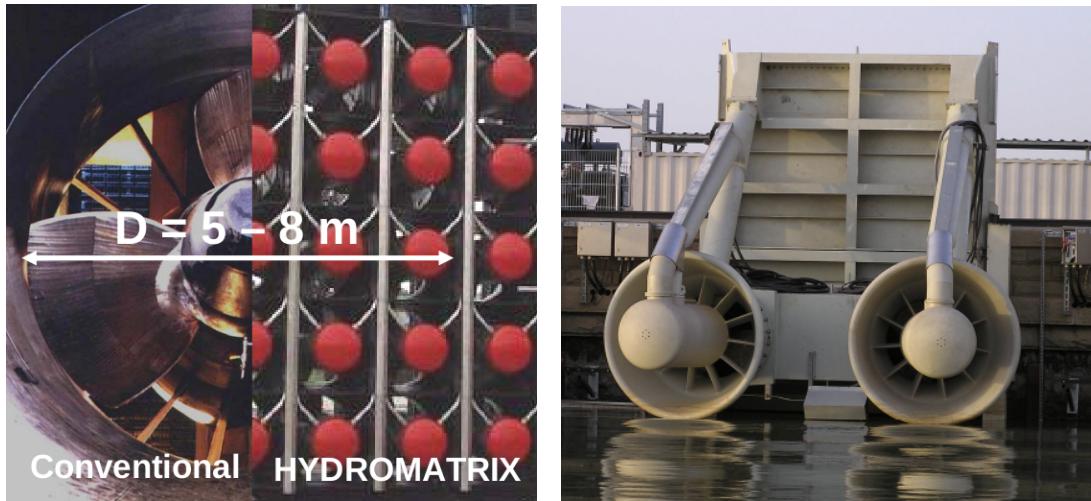


Figure 5.40: Optimization of a Hydromatrix® Turbine: Left: replacement of a single large turbine by a matrix of smaller ones. Right: Matrix turbines in raised position (in case of flood conditions, inspection or maintenance). From (3, 4).

Technical requirements regarding the economically feasible use of Hydromatrix® are listed below, (3, 4):

- Available plant discharge greater than $60m^3/s$.
- Available head from 2 to 30 m.
- Minimum submergence 1.5 m below tailwater.
- Unit output from 200 kW up to 700 kW.
- Close grid connection.
- Structures suitable for Hydromatrix®.
 - Navigation dams: Large lock and dam navigational structures along a number of major rivers.

5.4 Optimization of a Hydromatrix® Turbine

- Irrigation dams: Many structures for irrigation purposes exist worldwide, spilling water to agricultural areas on a regular basis.
- Sluice in shiplocks: Navigation river systems include dams and locks for ship transfer. In places where an existing slot is available, a Hydromatrix® module can be installed for power generation. The turbine-generator units can be designed to operate in both flow directions.
- Abandoned shiplocks: Due to the increasing navigation and transport activities on major rivers, many shiplocks have become too small and new shiplocks were built.

5.4.2 Case Presentation

This section is concerned with the design of a Hydromatrix® turbine for a given hydroelectric plant. The fact that Hydromatrix® is a new type of hydro turbine and that many untapped locations worldwide lay within their operating range denotes the importance of these design-optimization problems for the hydraulic turbine industry.

5.4.2.1 Case Formulation

The design problem in hand comprises the design of rotor blades, rotor hub and stay-vane end-tips. The distributor cone, the runner shroud and the draft tube are all fixed due to the modular construction and the generator size (fig. 5.41).

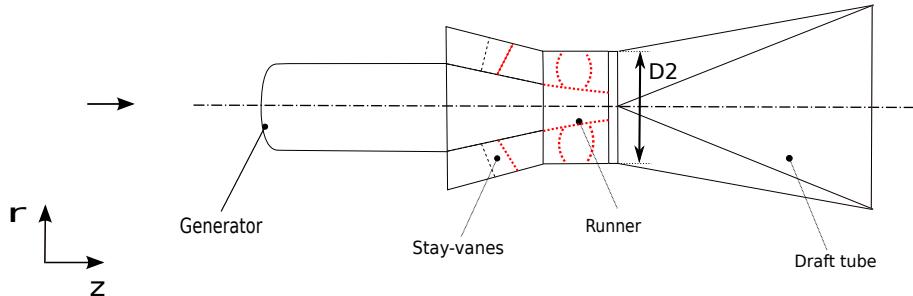


Figure 5.41: Optimization of a Hydromatrix® Turbine: Schematic representation, with red (light dotted) are the free to design parts and with black the fixed (due to construction constraints) ones.

5.4.2.2 Design Parameterization

In this case, the vector of design variables could be decomposed into two parts, a first one which is associated with the runner blade and the hub and a second one

5. Optimization of Hydraulic Turbomachines

which is related to the stay-vanes. The first part is based on the parameterization presented in section 5.1.1 and introduces 52 design variables to describe the rotor blade mean surface (table 5.7). The second part of the parameterization is associated solely with the stay-vane TE, since the stay-vanes thickness distributions are frozen for reasons related to their structural behaviour (stay-vanes must hold both the generator and runner weight in place). Also, the stay-vane LE is fixed since the flow meets the stay-vanes with a given axial velocity. The EA is concerned only with the first part of the parameterization. The second part is used to regulate the BE operating point as it will be shown later on.

Number of design variables	Design variables determining the:
6	Spanwise distributions of θ_{LE}
6	Spanwise distributions of θ_{TE}
6	Spanwise distributions of β_{LE}
6	Spanwise distributions of β_{TE}
6	Spanwise distributions of ζ_{LE}
6	Spanwise distributions of ζ_{TE}
8	LE projection on the meridional plane
8	TE projection on the meridional plane
52	Design variables, in total

Table 5.7: The 52 design variables used to parameterize the Hydromatrix® runner. This is the array of design variables the EA is dealing with.

For evaluating the performance of each runner geometry, the numerical simulation of the water flow through the turbine, based on the Euler equations, section 5.1.3, was used. The solver uses, as input, the rotational speed (n) and the total pressure drop from the inlet to the outlet (H). The flowrate Q is an outcome of the simulation. To get the desired Q value, required by the operation at the BE point, a procedure which iteratively deflects the stay-vane TE (second part of the parameterization) was used. This procedure is explained in detail in Appendix A.1. So, even if the optimization problem deals only with the design of the runner, the trailing part of the stay-vane iteratively adapts itself so as to fit the given rotor. The maximum number of iterations of the deflection angle α which were allowed was defined by the designer. This iterative process is the main reason that the CPU cost per evaluation may noticeably vary among the candidate solutions.

5.4 Optimization of a Hydromatrix® Turbine

5.4.2.3 Objectives and Constraints

The objectives vector consists of two objectives: f_1 is the outlet velocity profile metric (M_1) and f_2 is the combination of (a) the blade loading quality metric (M_2), (b) the cavitation index σ_i^{Hist} and (c) the pumping surface metric (M_3). The first objective is, therefore, concerned with the draft-tube coupling quality, seeking for blades with the minimal divergence from the user-defined target profiles shown in fig. 5.42. On the other hand, the second objective controls the pressure distribution over the runner blade, incorporating all relevant quality metrics. So

$$f_1^i = \alpha^i M_1^i \quad (5.16)$$

$$f_2^i = \beta^i M_2^i + \gamma^i \sigma_i^{Hist} + \delta^i M_3^i$$

	BE, $i=1$	PL, $i=2$	FL, $i=3$	Contributing to
$\alpha^i (M_1)$	1.0	0.0	0.0	f_1
$\beta^i (M_2)$	0.2	0.0	0.0	f_2
$\gamma^i (\sigma_i^{Hist})$	1.0	1.0	1.0	f_2
$\delta^i (M_3)$	0.0	100.0	100.0	f_2

Table 5.8: Optimization of a Hydromatrix® Turbine: Weights associated with the quality-metrics. Based on these weights, the quality metrics are grouped into two objective functions f_1 and f_2 .

5. Optimization of Hydraulic Turbomachines

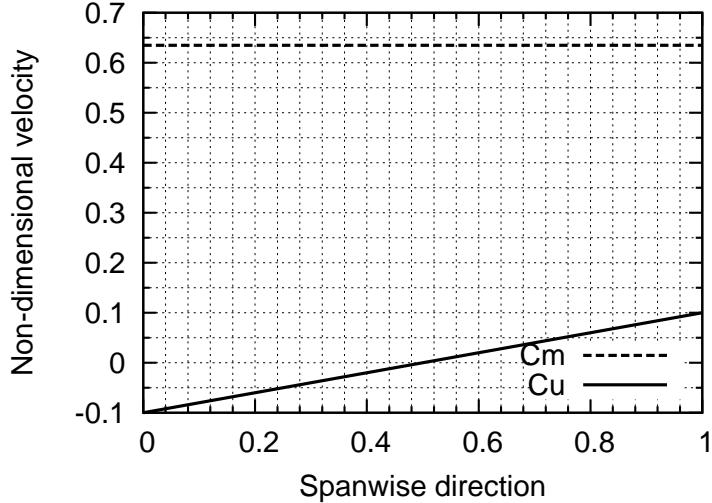


Figure 5.42: Optimization of a Hydromatrix® Turbine: Target non-dimensional velocity profile distributions at the runner outlet.

For the already defined $3 \times 2 = 6$ functions to be minimized at the three operating points, the two objective functions used in the optimization problem are defined by concatenating the previous ones with weight factors w_i (table 5.9), as follows

$$f_1 = \sum_{i=1}^3 w_i f_1^i \quad , \quad f_2 = \sum_{i=1}^3 w_i f_2^i \quad (5.17)$$

Operating point	Weight, w_i
Best efficiency (BE), $i=1$	1.0
Part-Load (PL), $i=2$	0.1
Full-Load (FL), $i=3$	0.1

Table 5.9: Optimization of a Hydromatrix® Turbine: Operating point weights.

5.4.3 Results

As mentioned in the beginning of this section, this case is used to demonstrate the advantages of the proposed use of PCA to drive the evolution operators, namely the so-called MAEA(PCA) method. Therefore, two optimization runs were performed using the conventional MAEA and the proposed MAEA(PCA).

5.4 Optimization of a Hydromatrix® Turbine

In both runs, the population sizes were $\mu=30$ and $\lambda=90$. In such a problem, the quite high value of the offspring population λ was really necessary since the blade shape parameterization and the bounds of the design variables may generate many infeasible solutions within the starting population. In more detail, for both runs, 88 out of the 90 individuals in the first generation corresponded to infeasible geometries and they all received death penalty. If all the initial generation members receive death penalty, then all of them get the same scalar cost function Φ and this can lead to premature evolution stagnation, constantly generating infeasible solutions. The inexact pre-evaluation was adjusted to start after the first 300 non-failed individuals entered the DB. Up to $\lambda_e=18$ top individuals per generation were allowed to undergo re-evaluation on the CFD tool. Local RBF networks were used as metamodels; each metamodel was trained on a few previously evaluated individuals in its vicinity. Based on the algorithm presented in (99), the designer defines the minimum (12) and maximum (16) number of training patterns to be used. It is important to note that, due to the very strict constraints, a great number of previously evaluated solutions which were given “death penalty” were recorded in the DB. The presence of so many infeasible entries in the DB can lead to a great number of infeasible entries included in the metamodels training set. This should be avoided since it hinders the training of dependable metamodels.

The application of the PCA-assisted evolution operators started at the 5th generation. In fig. 5.43, the performance of MAEA and MAEA(PCA) are compared using the hypervolume indicator.

MAEA(PCA) constantly finds a better front of non-dominated solutions during the entire course of evolution. From a different point of view, the quality of the front of non-dominated solutions obtained by MAEA after 2000 CFD-based evaluations could be obtained by the MAEA(PCA) at less than half this CPU cost.

The analysis of the non-dominated solution marked with *A*, fig. 5.44, follows. Fig. 5.45 shows the computed C_p iso-areas at BE point, over the entire turbine unit with its stator, rotor, hub and shroud.

The outlet flow quality of design A is presented in fig. 5.46, target distributions are achieved as desired. Furthermore the C_p profiles at the blade hub, mid-span and shroud locations, at the BE, PL and FL operating points are presented in figs. 5.48, 5.47 and 5.49, respectively. The small pumping surface, due to the propeller type of the turbine, inevitably makes its appearance at the PL point, as expected. It is though kept as small as possible. The pumping surface spans from the mid-span to the shroud as can be seen in fig. 5.47. In fact, a small pumping surface is allowed at the PL point given that this is a non-regulated machine.

The pumping surface phenomenon and its cause is shown in fig. 5.50. Pumping surface is the surface shown in dark blue color on the blades PS. This is caused

5. Optimization of Hydraulic Turbomachines

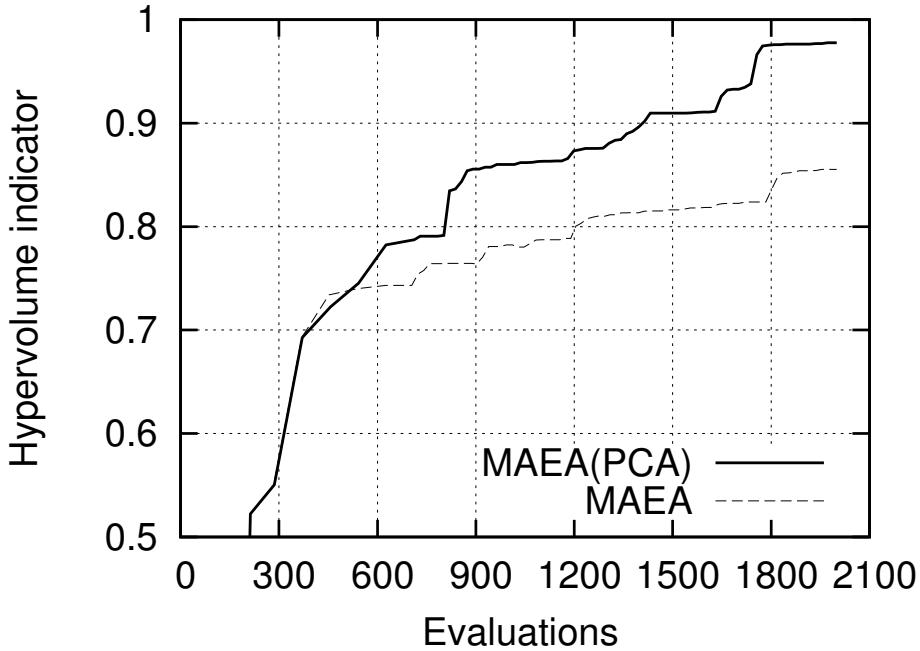


Figure 5.43: Optimization of a Hydromatrix® Turbine: The evolution of the hypervolume indicator for (a) the MAEA and (b) the MAEA(PCA) implementing the PCA-assisted evolution operators. It is evident that, since the PCA-assisted operators apply at the 5th generation, i.e. after about $4 \times 90 = 360$ evaluations based on the CFD tool, the first parts of the two curves are identical.

by the negative incidence angle, which means that the flow meets the blade at the SS just after the LE. This can be observed by comparing figs. 5.50, 5.52, and 5.51, for the PL, FL and BE operating points, respectively. The “best” incidence angle is associated with the BE operating point. The FL operating point shows a higher incidence angle than the BE, creating thus a lower pressure region near the LE at the SS, as can be seen by comparing figs. 5.48 and 5.49. As mentioned above, the PL operating point suffers from negative incidence angle, thus creating of higher pressure region near the LE at the SS (compare figs. 5.47 and 5.48), the so-called pumping surface phenomenon.

5.4 Optimization of a Hydromatrix® Turbine

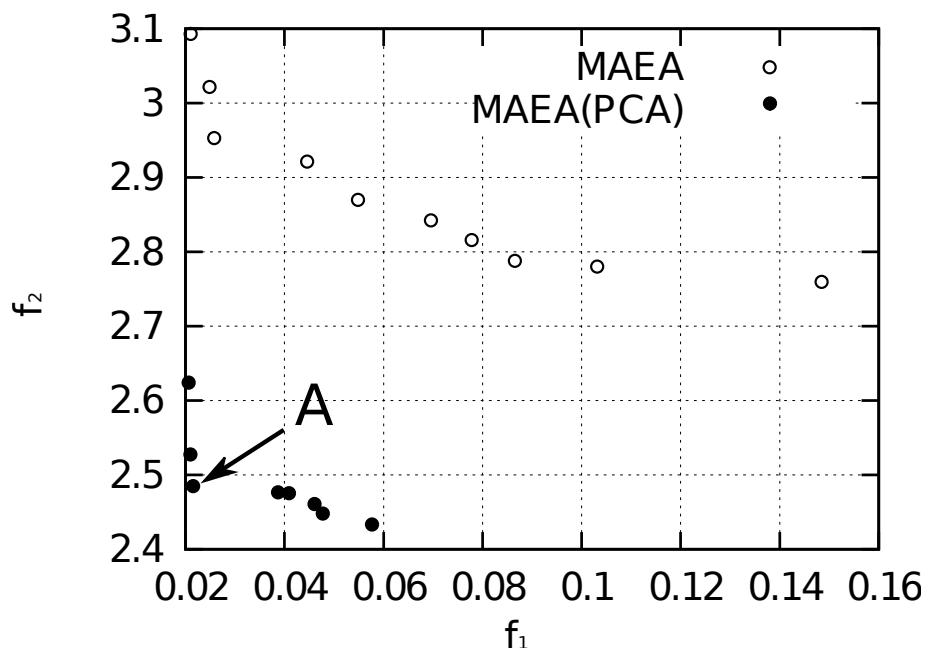


Figure 5.44: Optimization of a Hydromatrix® Turbine: Comparison of the fronts of non-dominated solutions computed using MAEA and MAEA(PCA), at the cost of 2000 CFD-based evaluations. Abscissa and ordinate correspond to the two objectives, as defined by eqs. 5.17.

5. Optimization of Hydraulic Turbomachines

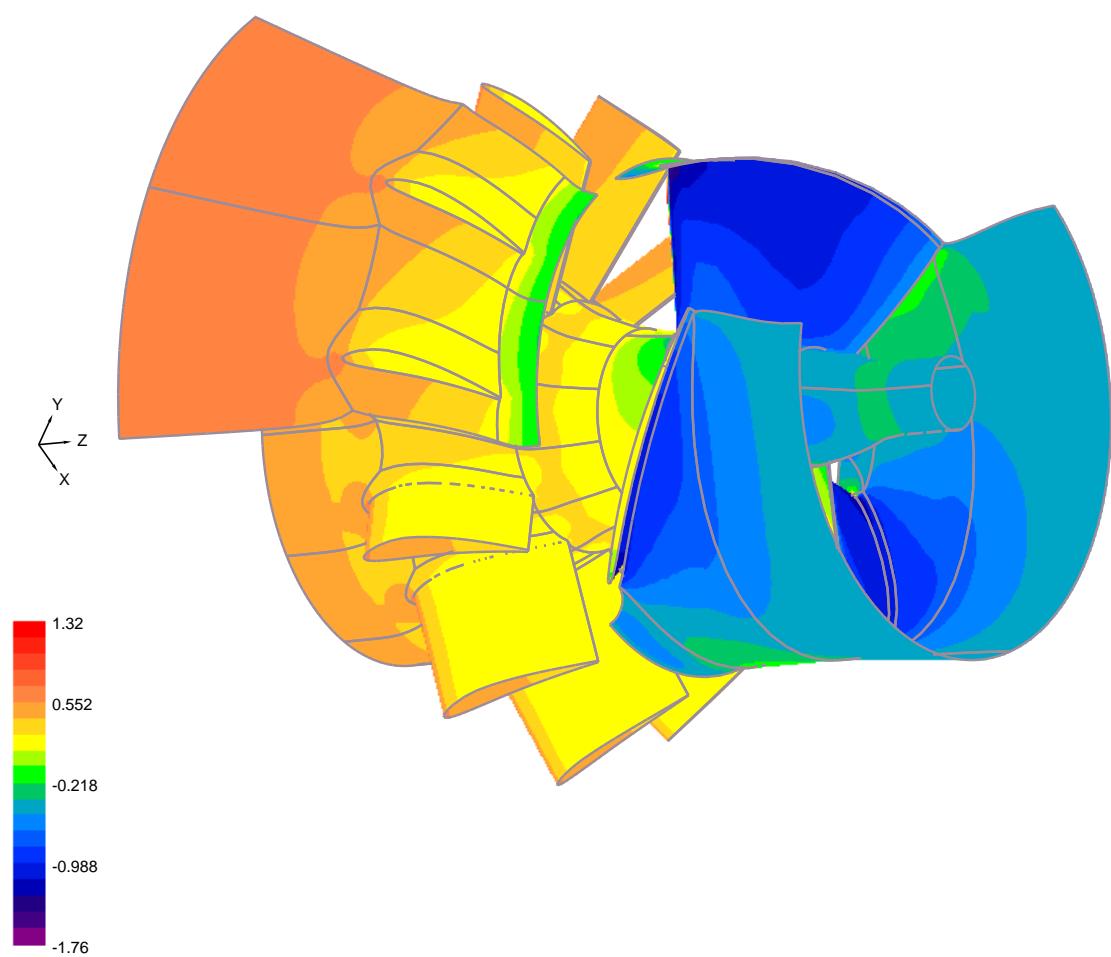


Figure 5.45: Optimization of a Hydromatrix® Turbine: C_p iso-areas for design A, at the BE point.

5.4 Optimization of a Hydromatrix® Turbine

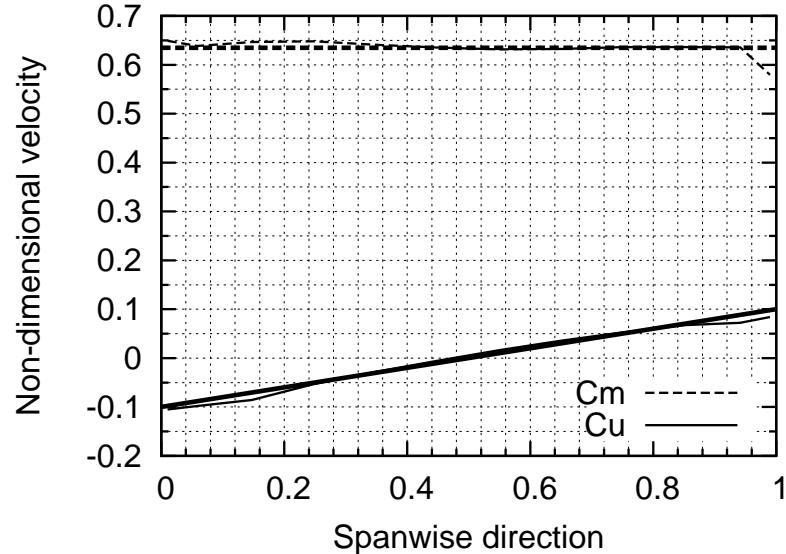


Figure 5.46: Optimization of a Hydromatrix® Turbine: meridional (C_m) and peripheral (C_u) outlet velocity profiles for design A, at the BE point.

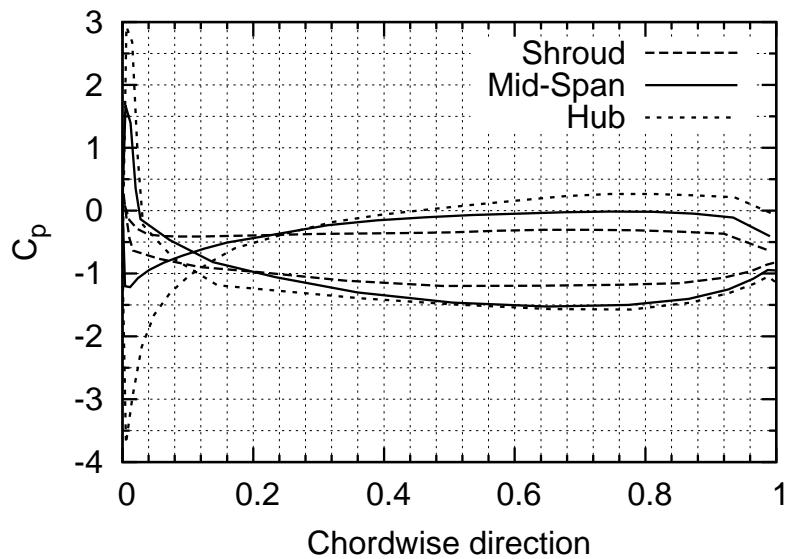


Figure 5.47: Optimization of a Hydromatrix® Turbine: C_p profiles for the hub, mid-span and shroud airfoils of design A operating at the PL point. The pressure overshooting near the LE, on the SS, due to the negative incidence angle (fig. 5.50), causes the so-called pumping surface phenomenon.

5. Optimization of Hydraulic Turbomachines

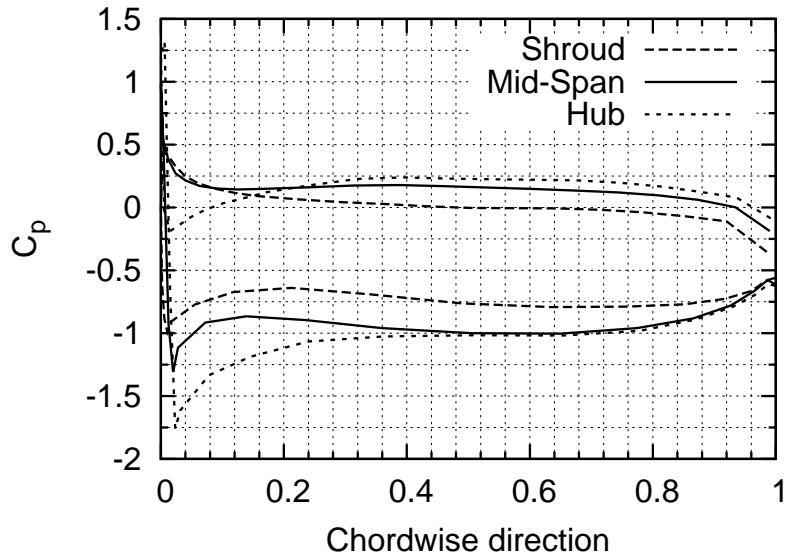


Figure 5.48: Optimization of a Hydromatrix® Turbine: C_p profiles for the hub, mid-span and shroud airfoils of design A operating at the BE point.

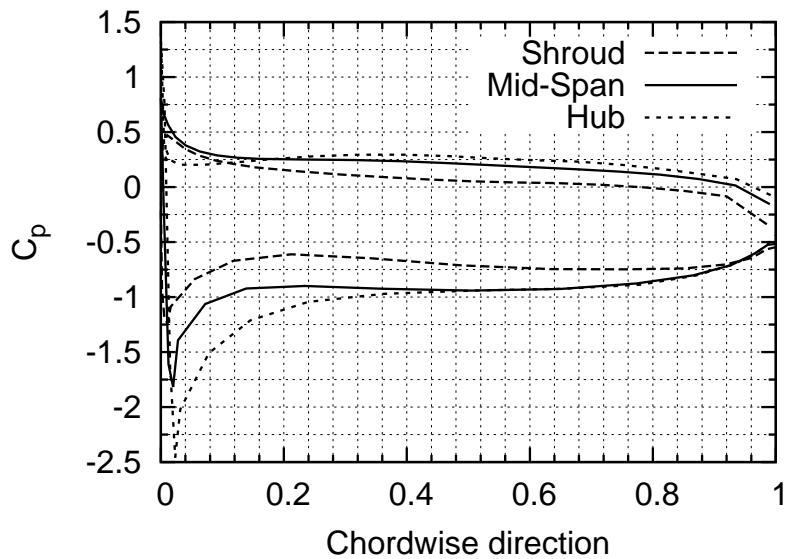


Figure 5.49: Optimization of a Hydromatrix® Turbine: C_p profiles for the hub, mid-span and shroud airfoils of design A operating at the FL operating point. Compared to its operation at the BE point, there is a pronounced undershooting close to the LE which can be explained by observing the flow incidence in fig. 5.52.

5.4 Optimization of a Hydromatrix® Turbine

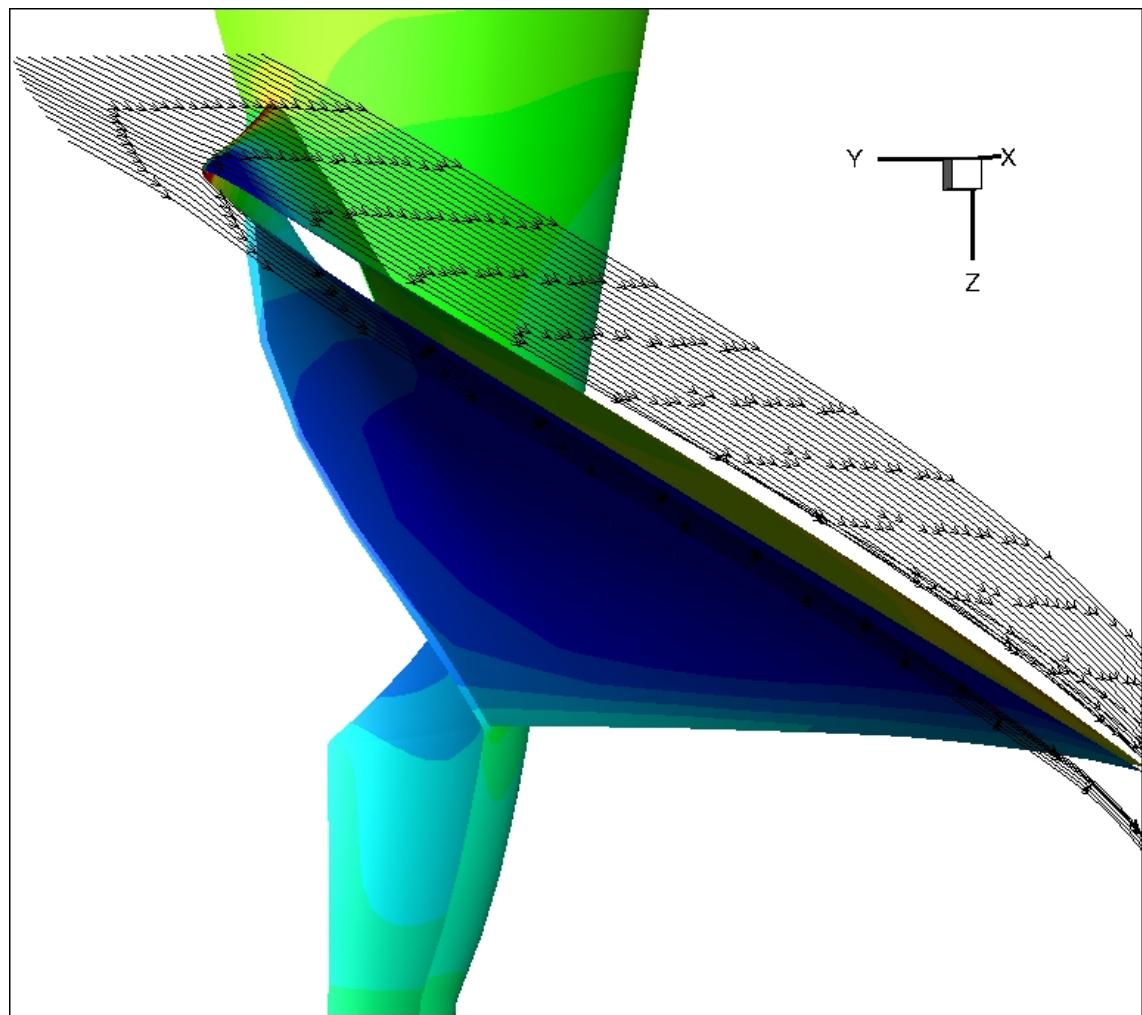


Figure 5.50: Optimization of a Hydromatrix® Turbine: Computed iso-bar areas for design A operating at the PL point. The water flow streamlines, near the shroud region, reveal the negative local incidence angle (the flow meets the blade at the SS), which causes the pumping surface phenomenon.

5. Optimization of Hydraulic Turbomachines

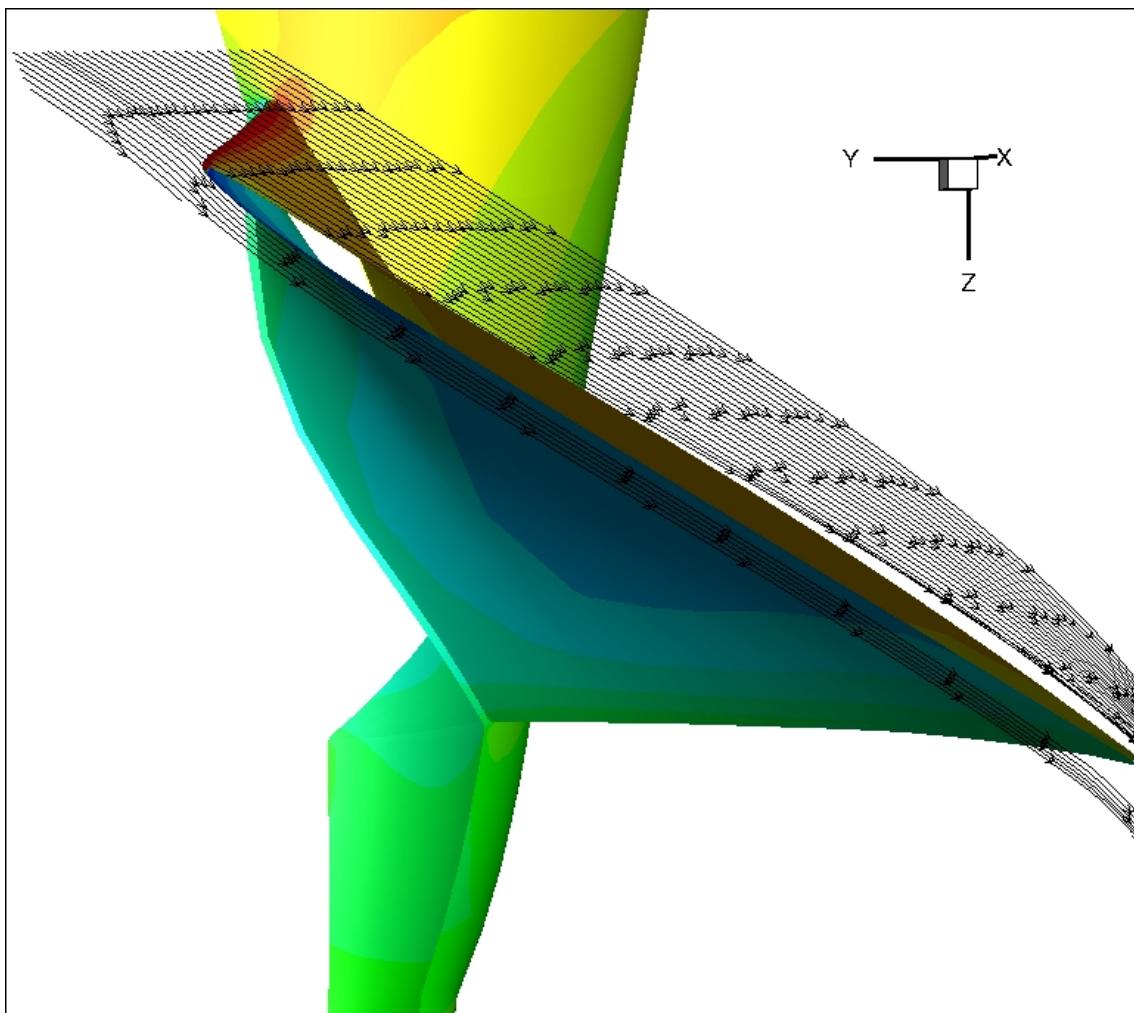


Figure 5.51: Optimization of a Hydromatrix® Turbine: Computed iso-bar areas for design A operating at the BE point. Water flow streamlines near the shroud region lead to “better” flow incidence angle, compared with the PL operation.

5.4 Optimization of a Hydromatrix® Turbine

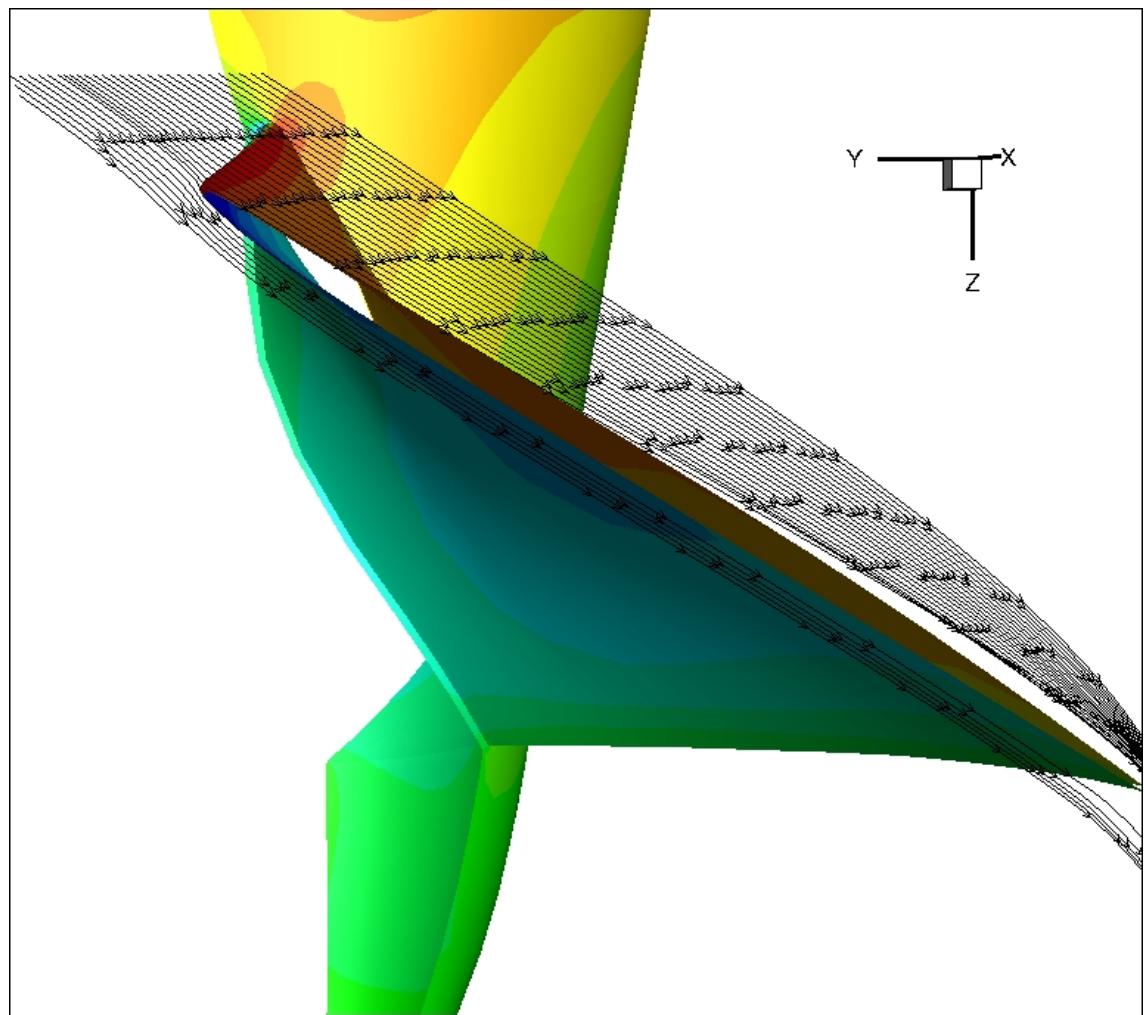


Figure 5.52: Optimization of a Hydromatrix® Turbine: Computed iso-bar areas for design A operating at the FL point. Water flow streamlines of the reveal a higher incidence angle than the BE, which causes a low pressure region near the LE at the SS, fig. 5.49.

5. Optimization of Hydraulic Turbomachines

6

Optimization of a Compressor Annular Cascade

In this chapter, the design-optimization of a stationary compressor annular cascade is presented. This case is used to assess the performance of the proposed M(PCA)AEA(PCA), as described in Chapter 4, and compare it with the MAEA developed in the framework of previous PhD theses (74, 92, 98) at PCOpt/LTT/NTUA. The same optimization problem was studied in (92) and is now revisited with the newly developed M(PCA)AEA(PCA).

The annular compressor annular cascade this chapter is dealing with has been installed at LTT/NTUA where both measurements (41, 125) and CFD computations (145) have been performed in the past. The cascade geometry was designed by SNECMA for investigating tip-clearance effects at high Mach numbers¹. This geometry is considered to be the reference one, throughout this chapter.

The evaluation software solves the Navier-Stokes equations for compressible flows through a time-marching, vertex-centered, finite volume method on unstructured hybrid grids. The aforementioned software was developed and validated at PCOpt/LTT/NTUA (15, 92). The convection terms are discretized via the Roe scheme (155). Second-order spatial accuracy is achieved through the MUSCL interpolation (180) along with the van Leer–van Albada limiting function (179).

The Navier-Stokes solver employed the high-Reynolds variant of the Spalart-Allmaras turbulence model (92). Since wall functions were used, the grid was coarse enough, in order to reduce as much as possible the CPU cost per evaluation. For all candidate blade shapes, grids of ~ 600.000 nodes were generated from scratch, using an automated grid generation tool (92). In general, any of these grids comprises ~ 225.000 tetrahedra, ~ 6.000 pyramids, ~ 200.000 prisms and

¹Experimental measurements on this cascade were performed in the framework of the project entitled “Advanced Civil Core Compressor Aerodynamics”, funded by the European Commission, AER2-CT92-0039, 1/1/1993-30/9/1996.

6. Optimization of a Compressor Annular Cascade

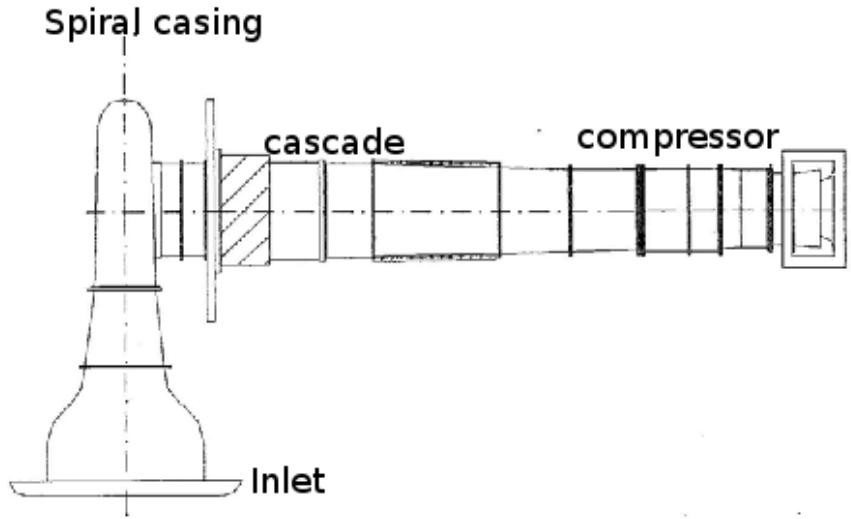


Figure 6.1: Optimization of a compressor annular cascade. Experimental set-up.

~ 400.000 hexahedrals. The average distance of the first nodes off the wall was $\sim 5 \times 10^{-5} m$ and the total time of grid generation was ~ 11 minutes on a modern single-core processor.

At the inlet, total pressure and temperature profiles along with the radial distributions of the peripheral and radial velocity flow angles were imposed as boundary conditions. At the exit, the static pressure at the hub was imposed and the pressure distribution over the exit plane was continuously updated by means of the radial equilibrium equation. Each CFD evaluation was carried out in parallel, by partitioning the unstructured grid into 8 equally-loaded sub-domains with the minimum interface between them so as to minimize the communication overhead. Running on an eight-core node the cost per CFD evaluation (excluding grid generation) is approximately 1.2 hours.

6.1 Blade Shape Parameterization

Since the blade airfoil shape is identical along the spanwise direction, the parameterization of a single airfoil is enough for the entire blade shape. Non-Uniform-Rational-B-Splines (NURBS) are used to control both airfoil sides. To parameterize the suction or the pressure side, 15 control points were used. Among them, 5 were free to change and, given that each control point has two degrees of freedom (its two coordinates), the number of design variables was 5×2 for the each side, yielding thus 20 degrees of freedom in total, (92).

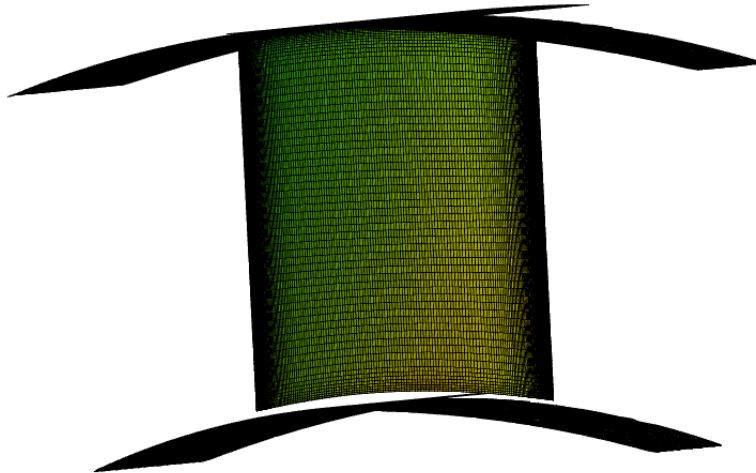


Figure 6.2: Optimization of a compressor annular cascade. The cascade blades are fixed at the shroud, forming the tip-clearance close to the stationary hub. Chord is equal to $C = 0.1m$ and tip-clearance height is equal to $0.02C$

6.2 Case Presentation

The results of the two different optimization runs, with the same 20 design variables each are presented in order to demonstrate the gains offered by the use of the methods proposed in Chapter 4. The first optimization run was carried out using the “standard” MAEA (i.e. the one presented in previous PhD theses carried out at PCOpt/LTT/NTUA, (92, 98)) with $\mu = 20$, $\lambda = 60$ and $\lambda_e = 4$. The IPE phase started once the first 220 individuals were stored in the DB. Metamodels, i.e. RBF networks, were trained on a number of already evaluated patterns which varied between 20 and 30; all training patterns were defined in \Re^{20} . The second optimization was based on the proposed M(PCA)AEA(PCA) method, with the same population sizes with the corresponding MAEA. In this run, RBF networks were trained on a much smaller number of training patterns (between 5 and 8) and all of the training patterns were defined in the \Re^{10} space. In this case, due to the reduced dimension, the IPE phase started earlier, just after the first 150 individuals were stored in the DB.

6.2.1 Objectives and Constraints

The optimization problem was solved with a single objective, i.e. aiming at the minimization of the spanwise-averaged pressure loss coefficient PLC_{av} . The PLC_{av} value results from the averaging (in spanwise direction) of the $PLC(r)$

6. Optimization of a Compressor Annular Cascade

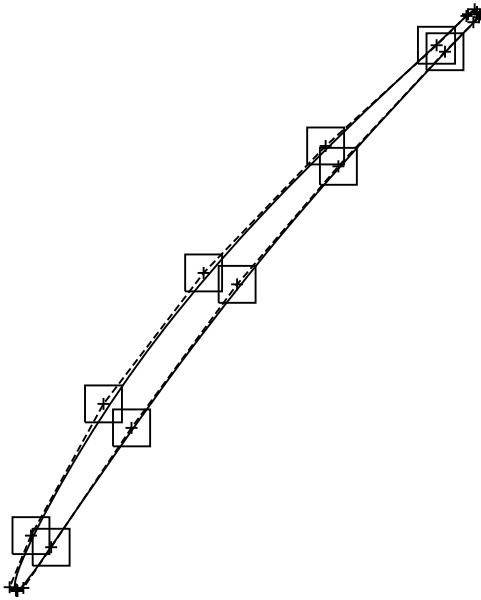


Figure 6.3: Optimization of a compressor annular cascade. The 20 design variables used for the design-optimization of the compressor annular cascade airfoil, (92). Boxes demonstrate the variable range for each design variable i.e. the space that each free control point may lay in. NURBS control point without a corresponding box are the (10 per side) fixed control points which are located at the leading and trailing edge region.

distribution (computed at each radial position). The latter is defined as

$$PLC(r) = \frac{p_{t,inl} - p_t(r)}{p_{t,inl} - p_{inl}} \quad (6.1)$$

Furthermore, the cascade to be designed must respect aerodynamic and geometrical constraints. The aerodynamic one requires that the minimum average outlet flow angle should not exceed 53° , namely $\alpha_2 < 53^\circ$. The reference blade of the compressor annular cascade installed at LTT/NTUA gives $\alpha_2 = 52.12^\circ$. Geometrical constraints were used to make the blade respect the desirable minimum airfoil thickness at three positions, namely at 30%, 60% and 90% of chord. The minimum allowed thickness values were set to the 90% of the corresponding reference values.

6.3 Results

Fig. 6.5 demonstrates the evolution of the objective function (PLC_{av}) values in terms of the number of evaluations for the MAEA and M(PCA)AEA(PCA). In fig. 6.4, the effects of the use of PCA during the metamodel training phase (earlier start and higher accuracy) is shown. The use of PCA during the application of the evolution operators further increases EA's efficiency (fig. 6.5). With the help of PCA, in order to guide both the metamodels and the evolution operators, the M(PCA)AEA(PCA) reached the PLC_{av} value of 0.104 at ~ 250 evaluations. MAEA needed ~ 500 evaluations to reach the same level of PLC_{av} .

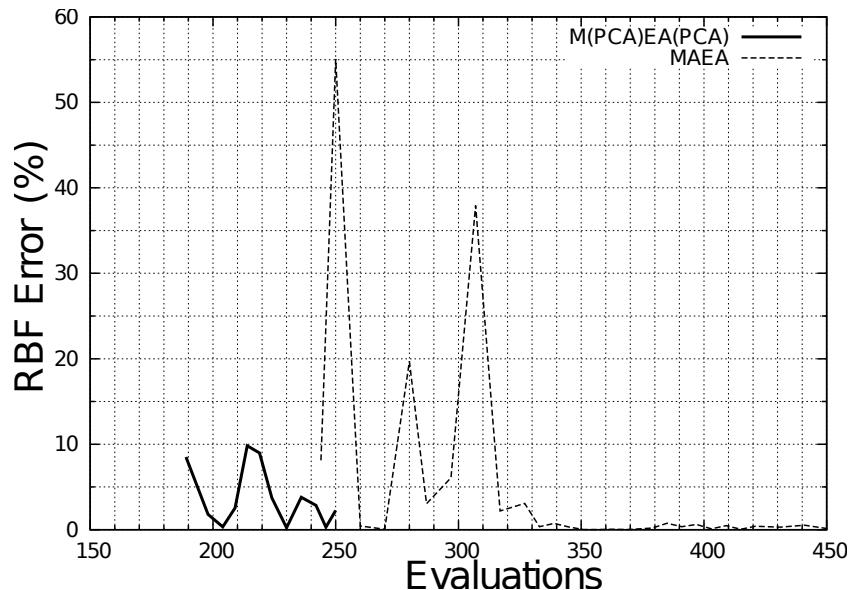


Figure 6.4: Optimization of a compressor annular cascade. Comparison of the metamodel (RBF) prediction error during the evolution, working either with MAEA or M(PCA)AEA(PCA). Since, in M(PCA)AEA(PCA), metamodels start being used earlier, the continuous line is shifted ahead.

The optimal airfoil shape computed using M(PCA)AEA(PCA) has $PLC_{av} = 0.104$ and is shown in fig. 6.6. This design respects all geometrical and aerodynamic constraints and its outlet flow angle is $\alpha_2 = 52.6^\circ$

6. Optimization of a Compressor Annular Cascade

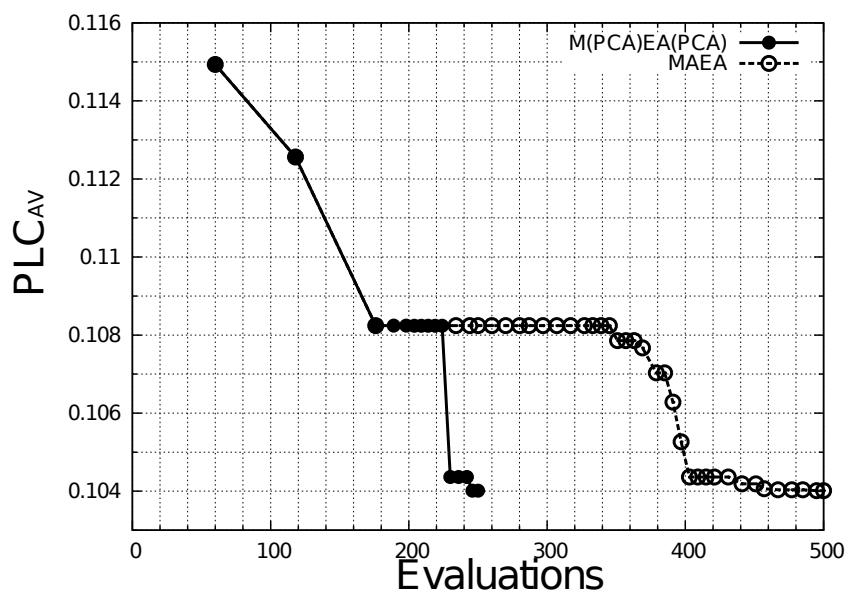


Figure 6.5: Optimization of a compressor annular cascade. Comparison of the performance of MAEA and M(PCA)AEA(PCA).

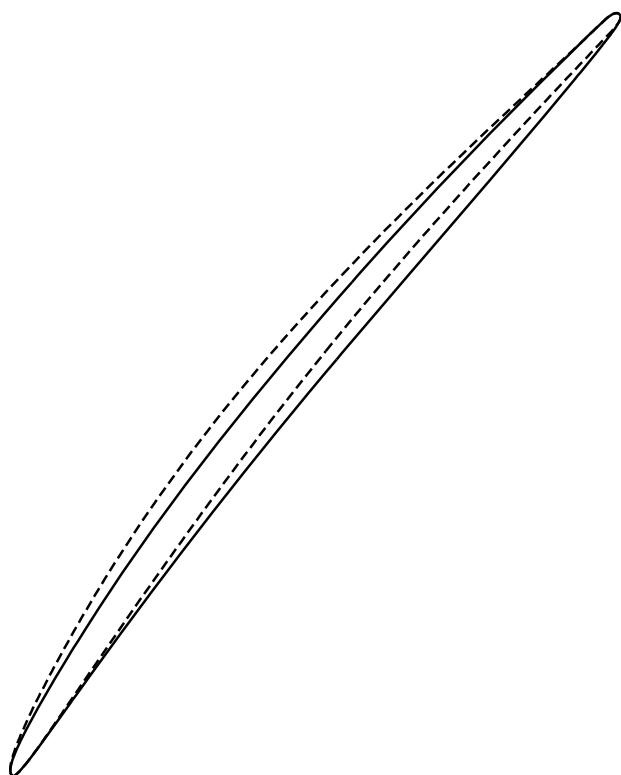


Figure 6.6: Optimization of a compressor annular cascade. The optimal blade airfoil shape computed using the M(PCA)AEA(PCA) (continuous line), compared with the reference one (dashed line).

6. Optimization of a Compressor Annular Cascade

7

Conclusions - Future Work

This PhD thesis aimed at proposing, developing, applying and validating new methods for upgrading Evolutionary Algorithms into design-optimization tools with reasonable computing cost, capable to undertake large-scale industrial optimization problems in the fields of thermal and hydraulic turbomachines. Throughout this PhD thesis, the proposed methods have been validated, in some low-cost mathematical optimization benchmarks, 2D compressor cascade designs and, finally, to the design of industrial 3D hydraulic turbines and a 3D compressor cascade with tip-clearance installed at LTT/NTUA.

The contribution of the thesis can be summarized in the three innovative methods listed below:

- (a) The Knowledge-Based Design (KBD) method. KBD is a new design method able to fully exploit a small number of available archived designs with good performance in “similar” conditions. Combining ideas from the theory of Knowledge-Based Systems (KBS) and EAs, KBD reduces significantly the optimization turn-around time. This is achieved by replacing the standard shape parameterization techniques, which often introduces a great number of design variables, using the available archived designs as design space basis vectors. In fact, KBD can be seen as a way to use the information residing into a small number of archived designs, made available from similar previous successful projects, so as to speed-up the optimization process.
- (b) The use of the Principal Component Analysis (PCA) of promising/top individuals, dynamically updated in each generation, in order to identify directions in the design space which are used to redefine the optimization variables resulting in “better-posed” optimization problems. Here, the redefinition of the optimization variables takes place by aligning each design vector with the so-called Principal Directions (PDs) computed by PCA. Then, the evolution operators are applied to the transformed optimization

7. Conclusions - Future Work

variables resulting in a more efficient search mechanism able to deal with initially “ill-posed” problems.

- (c) The use of PCA to make the use of metamodels in high-dimensional ill-posed optimization problems profitable. This is a way to further improve the MAEA efficiency by using the PCA of a small number of promising/top individuals in order to associate each design variable (or design space coordinate) with a degree of importance. It has been shown that MAEAs benefit a lot from the importance-based ranking of the design space coordinates, in order to overcome a well-known problem caused by the curse of dimensionality. This problem is related to the fact that for the ANNs used to approximate the cost of each new individual, any increase in the number of their sensory units makes the use of more training patterns mandatory and increases the training cost. Since the training patterns are selected among the previously evaluated individuals during the EA, the need for more training patterns means that the start of the IPE phase must be delayed. Additionally, each training becomes more costly, therefore, the gain from replacing the CFD code by metamodels is expected to be lower or almost zero. The proposed method is based on the selective truncation of the ANN sensory nodes, by maintaining only the most important ones according to the results of the PCA.

Using the proposed methods, the use of EAs to solve large-scale industrial applications (with a great number of design variables and non-separable objective functions) in acceptable by industry turn-around times became possible. In example, the overall design time (not only the optimization procedure) of a hydraulic turbine was reduced more than 50%, namely from about 120 to 140 days to only 50. This, in turn, reduced significantly the overall cost of the design procedure and, therefore, deemed the use of the designed models in sites with even lower hydraulic height, $H = 2, \dots, 5m$, and volume flow rates lower than the previously possible $60m^3/s$ economically feasible. The availability of sites with the aforementioned characteristics in Europe alone is approximately $6000GWh/year^1$.

The developed software is currently in use by Andritz-Hydro, one of the largest worldwide companies in the field of hydroelectric power plants. The company uses the developed software in a regular basis for the design of a wide range of rotating parts of hydraulic machines including runners for Francis, Kaplan, Bulb, Pump and Pump-Turbines. The developed software is also used by the same company for the design of static parts, such as draft tubes and distributors. A

¹ According to “European Commission, Joint Research Centre, Institute for Energy, Energy Systems Evaluation Unit, Petten, 13 of June 2007” in “Report on the Workshop on Hydropower and Ocean Energy – Part II: Hydropower”.

number of hydraulic machines, the design of which was based (among others) on the methods proposed in this thesis are currently installed or are ready to be installed, in various locations worldwide.

7.1 Future Work

Future research topics that could be considered as a sequel to this PhD thesis are:

- Investigation regarding the presence of non-linear variable correlations in ill-posed problems and ways to deal with them.
- The use of distributed EAs, based on clustering of the elite set and the computation of different set of PDs per deme according to its elite set.
- EAs for solving optimization problems that suffer from high fatality¹. Different sets of evolution operators could be used during the first stages of evolution in order to overcome this problem.
- Hybridization of EAs and gradient-based methods for use in the design of hydraulic turbines. This could take place through hierarchical search schemes using the adjoint variable method to compute the required gradients.
- Use ANNs in order to predict differences regarding the quality metrics between lower and higher levels of a hierarchical evaluation scheme and use them to appropriately update the target distributions of the lower level.
- The use of EAs with adaptive populations size would offer additional advantages. The population could be varied in order to keep quantities such as population-diversity constant.

7.2 Publications - Conference Presentations

Publications or conference presentations that took place during this PhD thesis are listed below:

S. Erne, M. Lenarcic, S.A. Kyriacou. Shape Optimization of a Flows Around Circular Diffuser in a Turbulent Incompressible Flow. ECCOMAS 2012 Congress, Vienna, Austria, September 10-14.

¹Problems where a great portion of the design space accommodates infeasible solutions.

7. Conclusions - Future Work

K.C.Giannakoglou, V.G. Asouti, S.A. Kyriacou, X.S. Trompoukis: ‘Hierarchical, Metamodel-Assisted Evolutionary Algorithms, with Industrial Applications’, von Karman Institute Lectures Series on ‘Introduction to Optimization and Multidisciplinary Design in Aeronautics and Turbomachinery’, May 7-11, 2012.

S.A. Kyriacou, S. Weissenberger and K.C. Giannakoglou. Design of a matrix hydraulic turbine using a metamodel-assisted evolutionary algorithm with PCA-driven evolution operators. International Journal of Mathematical Modelling and Numerical Optimization, SI: Simulation-Based Optimization Techniques for Computationally Expensive Engineering Design Problems, 3(2):45–63, 2012.

I.A. Skouteropoulou, S.A. Kyriacou, V.G Asouti, K.C. Giannakoglou, S. Weissenberger, P. Grafenberger. Design of a Hydromatrix turbine runner using an Asynchronous Algorithm on a Multi-Processor Platform. 7th GRACM International Congress on Computational Mechanics, Athens, 30 June-2 July, 2011.

S.A. Kyriacou, S. Weissenberger, P. Grafenberger, K.C. Giannakoglou. Optimization of hydraulic machinery by exploiting previous successful designs. 25th IAHR Symposium on Hydraulic Machinery and Systems, Timisoara, Romania, September 20-24, 2010.

H.A. Georgopoulou, S.A. Kyriacou, K.C. Giannakoglou, P. Grafenberger and E. Parkinson. Constrained multi-objective design optimization of hydraulic components using a hierarchical metamodel assisted evolutionary algorithm. Part 1: Theory. 24th IAHR Symposium on Hydraulic Machinery and Systems, Foz do Iguassu, Brazil, October 27-31, 2008.

P. Grafenberger, E. Parkinson, H.A. Georgopoulou, S.A. Kyriacou and K.C. Giannakoglou. Constrained multi-objective design optimization of hydraulic components using a hierarchical metamodel assisted evolutionary algorithm. Part 2: Applications. 24th IAHR Symposium on Hydraulic Machinery and Systems, Foz do Iguassu, Brazil, October 27-31, 2008.

Appendix A

Regulated Hydraulic Turbines

In order to increase the operational range of a hydraulic turbine in terms of allowed flowrates Q and hydraulic heads H , the concept of regulation is frequently used. A regulated hydraulic turbine is a turbine in which the stator or rotor or both have adjustable blades. An adjustable blade can rotate around a predefined axis by means of an appropriate mechanism.

A.1 Single-Regulated Turbines

Hydraulic turbines that have a single row of adjustable blades (usually the stator blades) are called single-regulated turbines (fig. A.1). Typical examples of single-regulated turbines are Francis turbines, axial fixed-blade propeller turbines with adjustable stator blades and the so-called semi-Kaplan concept with fixed stator and adjustable rotor blades, (43, 140). Though the rotation angle α^j of the adjustable blades might be handled as one additional design variable per operating point j , a different approach is proposed in this thesis.

Regarding its application to single-regulated turbines, the proposed method has the advantage of: (a) removing the burden of computing the appropriate α^j angles by the EA and (b) eliminating the need to impose additional constraints in order to ensure operation at the desirable operating points. Instead, through an iterative procedure, it is sufficient to ensure that each candidate solution computes the appropriate α^j angles for operating at the desirable points. Quality metrics are then computed and the evolution continues as presented in Chapter 5. To do so, evaluation procedure is modified by including the computation of the right α^j value for each operating point j . In specific, for each candidate solution and for each operating point:

Step 1: (Initialization)

A. Regulated Hydraulic Turbines

Step 1a: (Initialize α) Sets the iteration counter to $i = 0$ and $\alpha_i = \alpha_0$, where α_0 a user-defined value.

Step 1b: (First Step) Sets $i = 1$ and $\Delta\alpha = \Delta\alpha_0$, where $\Delta\alpha_0$ the user-defined increment and defines

$$\alpha_1 = \begin{cases} \alpha_0 - \Delta\alpha_0 & , \text{if } (Q < Q_{required}) \text{ or } (H > H_{required}) \\ \alpha_0 + \Delta\alpha_0 & , \text{if } (Q > Q_{required}) \text{ or } (H < H_{required}) \end{cases} \quad (\text{A.1})$$

By definition, $\alpha = 0^\circ$ corresponds to the fully open stator position.

Step 2: (Gradient Computation) Sets $i = i + 1$. Computes derivatives $\frac{\partial(Q - Q_{required})}{\partial\alpha}$ or $\frac{\partial(H - H_{required})}{\partial\alpha}$, depending on the boundary conditions employed in the flow solver. Finite-difference quotients such as the following

$$\frac{\partial(Q - Q_{required})}{\partial\alpha} = \frac{(Q_{i-1} - Q_{required}) - (Q_{i-2} - Q_{required})}{\alpha_{i-1} - \alpha_{i-2}} \quad (\text{A.2})$$

are used.

Step 3: (Update α) Based on the Newton-Raphson method, updates α as follows

$$\alpha_i = \alpha_{i-1} - \eta \frac{Q_{i-1} - Q_{required}}{\frac{\partial(Q - Q_{required})}{\partial\alpha}} \quad (\text{A.3})$$

where η is a user-defined relaxation factor.

Step 3: (Convergence Check) Stops if $|Q_i - Q_{required}| < \Delta Q^*$, where ΔQ^* a user-defined threshold value; otherwise, continues from step 2.

This algorithm can be scaled to any number of operating points, in case of multi-operating point design problems.

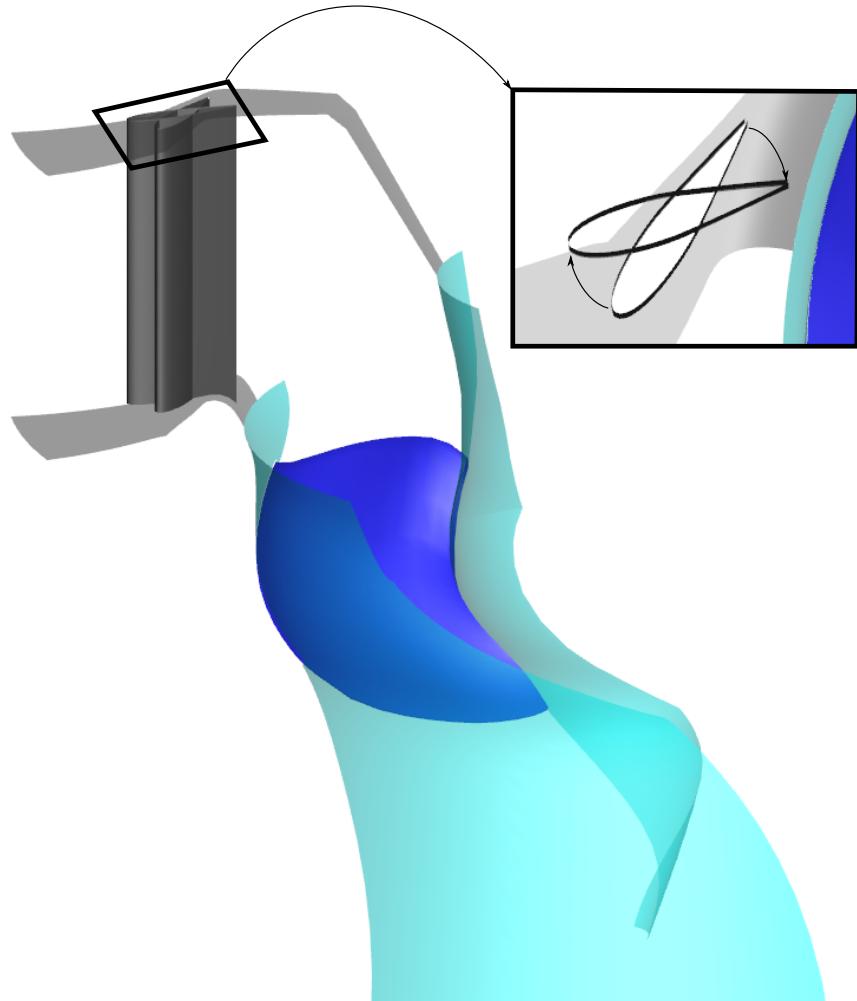


Figure A.1: Single-regulated (axial) turbine. The adjustable stator blades are shown in grey whereas the non-adjustable rotor blades in blue.

A.2 Double-Regulated Turbines

Hydraulic turbines that have two rows of adjustable blades (rotor and stator, fig. A.2) are called double-regulated turbines, (43, 140). A typical example is the Kaplan turbine. In double-regulated turbines, the stator rotation angle is denoted by α (in accordance with what was used before) and the rotor rotation angle by β .

To handle the design of double-regulated turbines, the algorithm proposed for the single-regulated ones to compute the α^j angles is used. Over and above, the introduction of β^j , where j is the operating point counter, as additional design variables takes place. Doing so, the EA overcomes the computation of the α^j

A. Regulated Hydraulic Turbines

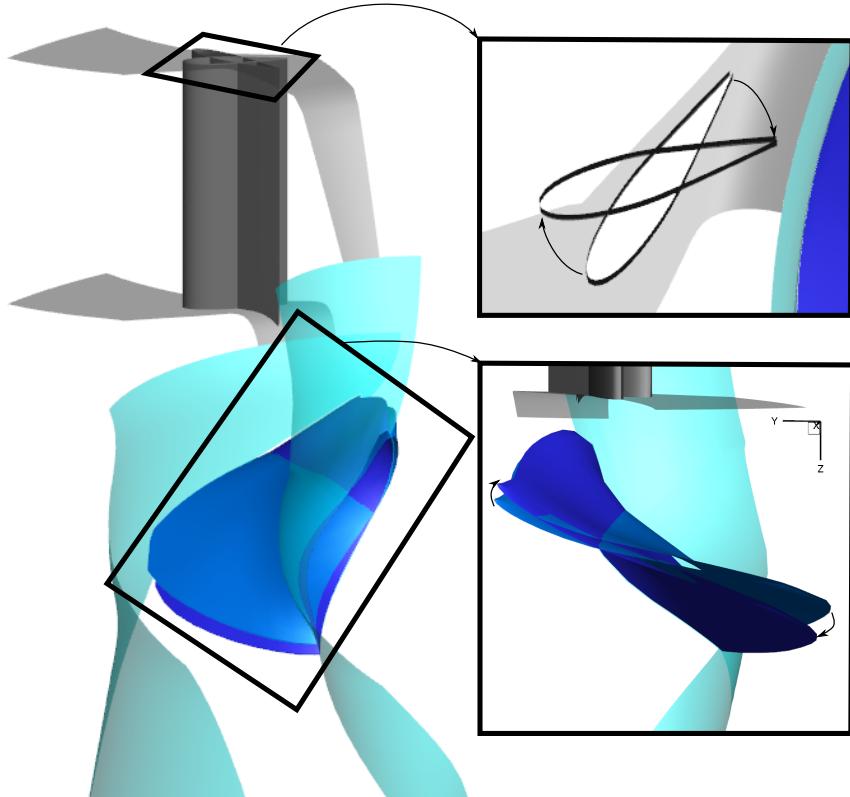


Figure A.2: Double-regulated (Kaplan) turbine. In contrast to fig. A.1, the rotor blades (in blue) are also adjustable.

angles (as for single-regulated turbines) and undertakes only the search for the optimal β^j angles so as to achieve best performance at each operating point.

Appendix B

Design of a Hydromatrix®: Experimental Validation

In this Appendix, the experimental validation of a Hydromatrix® design, which was carried out using the methods presented in this thesis, is presented. This design was associated with the EU funded project “HYDROACTION – Development and laboratory testing of improved action and Matrix hydro turbines designed by advanced analysis and optimization tools” (FP7: Project Number 211983) in which both NTUA and Andritz-Hydro participated.

The design shown below refers to a Hydromatrix® hydraulic turbine which must operate optimally at the three operating points of table B.1. The optimization procedure used to generate the design under consideration is presented first. Then, the experimental validation of the quality of the final design in terms of efficiency is presented and the new design is compared with an existing/reference one.

B.1 Optimization Procedure

The design variables are in accordance with the parameterization technique presented in Chapter 5. There are 52 design variables in total, used to describe the rotor blade mean surface, its thickness distribution and the hub generatrix (table 5.7).

Three objectives are used: f_1 is the outlet velocity profile metric M_1 , f_2 is the combination of the blade loading quality metric M_2 and the cavitation index σ^{Hist} cast in a single objective and, finally, f_3 coincides with the pumping surface metric M_3 . By considering the three operating points, three objectives (f_1^i, f_2^i, f_3^i) for each operating point i are defined as follows,

B. Design of a Hydromatrix®: Experimental Validation

Operating point	N_{ED}	Q_{ED}
BE	0.97	0.62
PL	1.42	0.84
FL	0.92	0.6

Table B.1: Experimental validation of the design of a Hydromatrix®: The three operating points.

$$f_1^i = \alpha^i M_1^i \quad , \quad f_2^i = \beta^i M_2^i + \gamma^i \sigma_i^{Hist} \quad , \quad f_3^i = \delta^i M_3^i \quad (\text{B.1})$$

where α^i , β^i , γ^i and δ^i are given in table B.2

	BE, $i=1$	PL, $i=2$	FL, $i=3$	Associated objective
$\alpha^i (M_1)$	1.0	1.0	1.0	f_1
$\beta^i (M_2)$	0.2	0.2	0.2	f_2
$\gamma^i (\sigma_i^{Hist})$	1.0	1.0	1.0	f_2
$\delta^i (M_3)$	0.0	100.0	100.0	f_3

Table B.2: Experimental validation of the design of a Hydromatrix®: Weights associated with the grouping of quality-metrics into objectives.

Given the 3×3=9 functions f_j^i to be minimized, the three objective functions handled by the EA are defined by multiplying the previous ones with weight factors w_i (table B.3) and summing them up as follows

$$f_1 = \sum_{i=1}^3 w_i f_1^i \quad , \quad f_2 = \sum_{i=1}^3 w_i f_2^i \quad , \quad f_3 = \sum_{i=1}^3 w_i f_3^i \quad (\text{B.2})$$

Operating point	Weight, w_i
Best efficiency (BE), $i=1$	1.0
Part load (PL), $i=2$	0.1
Full load (FL), $i=3$	0.1

Table B.3: Experimental validation of the design of a Hydromatrix®: Operating point weights.

The resulting front of non-dominated solutions computed at the cost of 2000

B.1 Optimization Procedure

evaluations is presented in fig. B.1. Upgrading the cavitation index σ^{Hist} from constraint to objective allowed, at the cost of a single optimization run, to compute a number of designs, each of them being optimal regarding different cavitation safety requirements. It is reminded that a design can safely operate with higher σ^{Hist} , without suffering from cavitation, if placed at a greater depth. Also, the Hydromatrix® turbines are usually placed in rows, one above the other, therefore the use of σ^{Hist} as an objective gives the ability, with just a single run, to have all turbines needed for the entire project designed.

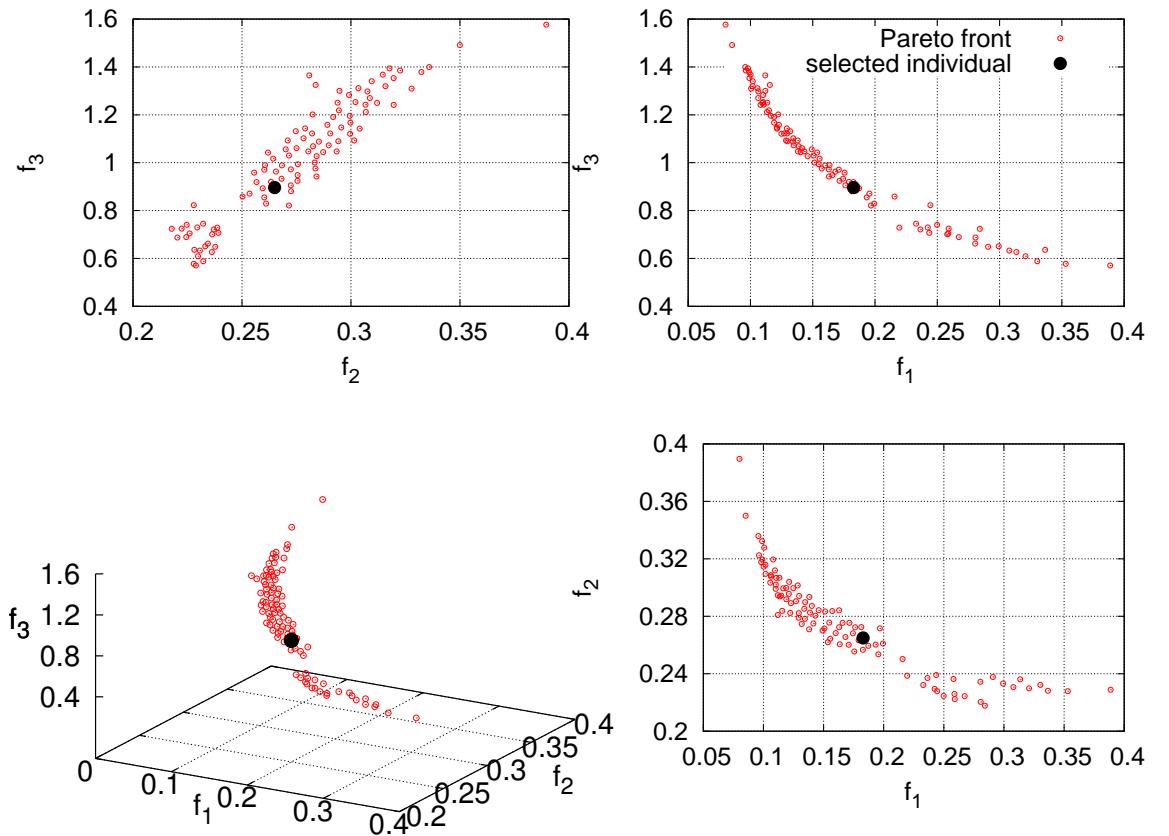


Figure B.1: Experimental validation of the design of a Hydromatrix®: The 3D front of non-dominated solutions computed at the cost of 2000 evaluations and the design selected to undergo construction and experimental validation. The 3D plot and its projections onto the $((f_2, f_3), (f_1, f_3)$ and (f_1, f_2) planes are shown.

For the experimental validation, a single design from the computed front of non-dominated solutions was selected, manufactured and measured in test rig L2b of Andritz Hydro GmbH in Linz, Austria (fig. B.4). The selected design is compared, efficiency-wise, with a reference Hydromatrix® turbine designed to operate at a “similar” operating region.

B. Design of a Hydromatrix®: Experimental Validation

Before proceeding to the experimental validation, an analysis of the numerical simulations performed for the selected design follows. In fig. B.2 (left), the high loading quality of the selected design is demonstrated through the C_p chordwise distribution (eq. 5.8) along the hub, mid-span and shroud. In fig. B.2 (right), the excellent agreement of the outlet velocity profiles (C_m and C_u) with the desired ones is presented. Furthermore, in fig. B.3, the C_p distributions demonstrate the loading quality of the selected design when it operates at the FL and PL operating points.

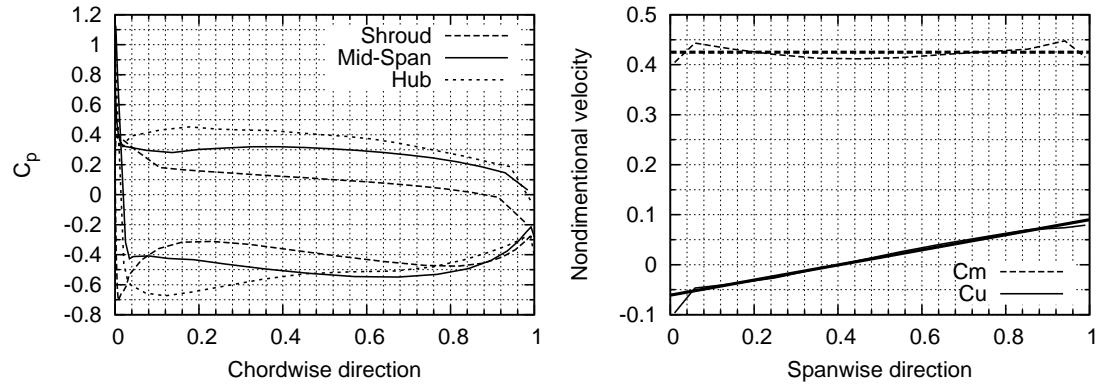


Figure B.2: Experimental validation of the design of a Hydromatrix®: Left: Computed C_p profiles at hub, mid-span and shroud for the selected design operating at the BE point. Right: Computed C_m and C_u outlet velocity profiles for the selected design, at the BE point.

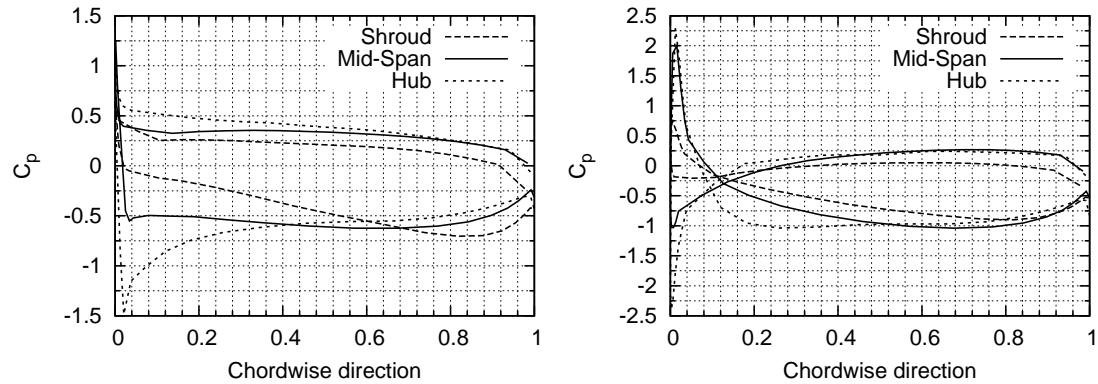


Figure B.3: Experimental validation of the design of a Hydromatrix®: Computed C_p profiles at hub, mid-span and shroud for the selected design operating at the FL point. Right: Computed C_p profiles at hub, mid-span and shroud for the same design, at the PL point

B.2 Experimental Validation

The experimental validation of the quality of the selected Hydromatrix® design was performed at the test rig L2b (fig. B.4) which is specialized for Bulb, Straflo and Hydromatrix® turbines. In L2b, three pumps are used to simulate the desirable river flow (Q , H). The model turbine is installed between head- and tail-water tanks.

The model head is regulated by the pump speed and the tail water elevation by the pressure in the downstream tank. The latter is connected to a vacuum vessel and compressed air. This allows varying the downstream tank pressure for the simulation of the necessary cavitation conditions, while keeping the model head constant. For the Hydromatrix® tests, a special inlet box to simulate the river flow is required.

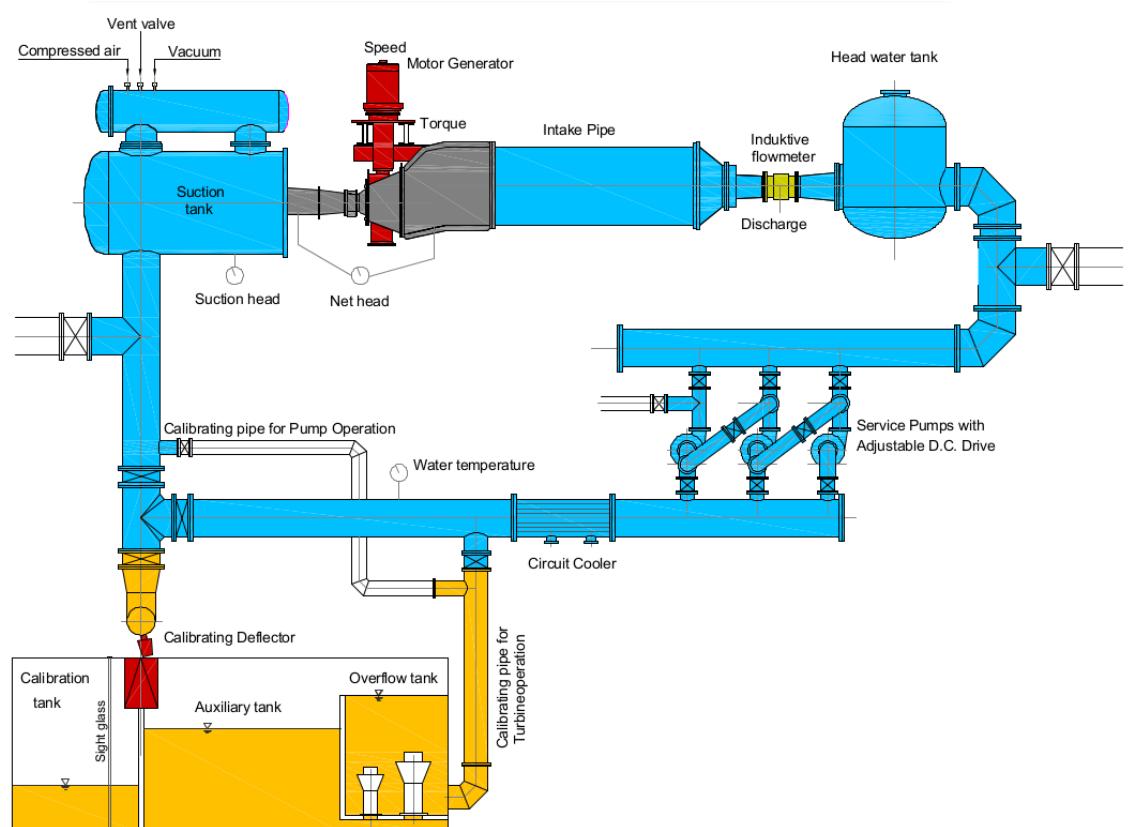


Figure B.4: Experimental validation of the design of a Hydromatrix®: General layout of test rig L2b of Andritz-Hydro/Linz in the Hydromatrix® configuration. L2b is operated as a closed loop and is able to perform a variety of tests in accordance with international standards “IEC 60193” (173).

B. Design of a Hydromatrix®: Experimental Validation

Regarding the torque measurement, a strain-gage equipped shaft, with its signal being transmitted via a wireless telemetry system to the data acquisition system is used. The model head (H) is computed from the static pressure difference between the head-water and tail-water tanks and the velocity difference between the measurement sections (distributor cone inlet and draft tube outlet). Volume flow rate Q is measured using an inductive discharge flowmeter; rotational speed is measured using an incremental encoder with a resolution of 1024 impulses per revolution. All measuring instruments were calibrated either by primary normals or by using calibration devices which are calibrated with a primary normal. The runner blade geometry was checked with a 3D coordinate measurement machine by an external company.

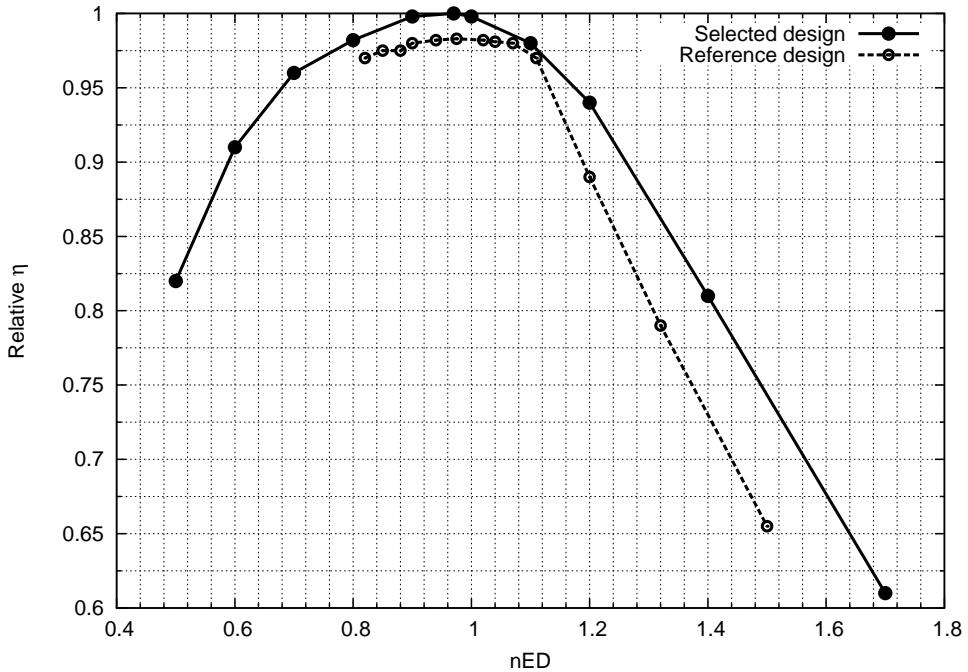


Figure B.5: Experimental validation of the design of a Hydromatrix®: Relative efficiency ($\frac{\eta}{\max(\eta)}$) charts for the reference and the selected design. The selected design, optimized using the proposed methods, outperforms the reference one throughout the operating spectrum.

Efficiency (η) measurements were performed at the minimum model Reynolds number of 4×10^6 and minimum specific hydraulic energy of $30 J/kg$, as indicated in IEC 60193 (173). In order to acquire the efficiency curve, head and speed were changing so as to make the speed factor N_{ED} vary too. The measured efficiency of the selected design is shown, in comparison with a reference design, in fig. B.5.

The experimental validation showed that the methods proposed in this thesis

B.2 Experimental Validation

deliver high quality designs, acceptable by industrial standards. Furthermore, the overall design time reduced from approximately 120 – 140 days to only 50. This reduction in time and, therefore, resources, combined with the ability to treat turbine parts, such as the draft tube, as standardised components (merely requiring their coupling properties) resulted in the reduction of the overall cost per unit to, approximately, one third.

This cost reduction makes the use of this type of turbines, in power plants with even lower H (below 5m, as low as 2m) and relatively small (smaller than $60m^3/s$) discharges, viable. The economic impact of this improvement can be very important as the current estimations for the potential of sites in the range of 2 to 5m is approximately 6000 GWh per year in Europe alone¹, a market worth potentially more than a billion Euro per year.

¹This figure was given by “European Commission, Joint Research Centre, Institute for Energy, Energy Systems Evaluation Unit, Petten, 13 of June 2007” in “Report on the Workshop on Hydropower and Ocean Energy – Part II: Hydropower”.

B. Design of a Hydromatrix®: Experimental Validation

Bibliography

- [1] <http://acoustic-interfaces.at/sites/aigrid.htm>.
- [2] <http://www.andritz.com/hydro.htm>.
- [3] <http://www.andritz.com/ANONID2C30C9DB22C0E151/hydro-products-and-services-large-hydro-hydromatrix>.
- [4] <http://hydromatrix.info/index.htm>.
- [5] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. pages 39–59. AI Communications, 7(i), 1994.
- [6] R. Akerkar and P. Sajja. *Knowledge-Based Systems*. Jones and Bartlett Publishers Inc., 1st edition, 2009.
- [7] E. Alba and B. Dorronsoro. *Cellular Genetic Algorithms*. Springer-Verlag, 2008.
- [8] E. Alba and J.M. Troya. Analyzing synchronous and asynchronous parallel distributed genetic algorithms. *Future Generation Computer Systems*, 17(4):451–465, 2001.
- [9] J.S. Anagnostopoulos and D. E. Papantonis. Optimal sizing of a run-of-river small hydropower plant. *Energy Conversion and Management*, 48:2663–2670, 2007.
- [10] J.S. Anagnostopoulos and D.E. Papantonis. Flow Modeling and Runner Design Optimization in Turgo Water Turbines. *International Journal of Applied Science, Engineering and Technology*, 4(3):136–141, 2007.
- [11] J.S. Anagnostopoulos and D.E. Papantonis. Pumping station design for a pumped-storage wind-hydro power plant. *Energy Conversion and Management*, 48:3009–3017, 2007.

BIBLIOGRAPHY

- [12] J.S. Anagnostopoulos and D.E. Papantonis. Simulation and size optimization of a pumped–storage power plant for the recovery of wind-farms rejected energy. *Renewable Energy*, 33:1685–1694, 2008.
- [13] W.K. Anderson and D.L. Bonhaus. Aerodynamic design on unstructured grids for turbulent flows. *AIAA Journal*, 37(2):185–191, 1999.
- [14] W.K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers and Fluids*, 28:443–480, 1999.
- [15] V.G. Asouti. *Μέθοδοι αεροδυναμικής ανάλυσης και σχεδιασμού για ροές υψηλών και χαμηλών ταχυτήτων, σε πολυεπεξεργαστικό περιβάλλον*. PhD thesis, National Technical University of Athens, 2009.
- [16] V.G. Asouti and K.C. Giannakoglou. Aerodynamic optimization using a parallel asynchronous evolutionary algorithm controlled by strongly interacting demes. *Engineering Optimization*, 41(3):241–257, 2009.
- [17] V.G. Asouti, I.C. Kampolis, and K.C. Giannakoglou. A Grid-Enabled Asynchronous Metamodel-Assisted Evolutionary Algorithm for Aerodynamic Optimization. *Genetic Programming and Evolvable Machines (SI:Parallel and Distributed Evolutionary Algorithms, Part One)*, 10(3):373–389, 2009.
- [18] S. Axler. *Linear Algebra Done Right*. Springer Verlag, 1997.
- [19] T. Back. In *Evolutionary Algorithms in Theory and Practice, evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, 1996.
- [20] N. Benoudjit, C. Archambeau, A. Lendasse, J. Lee, and M. Verleysen. Width optimization of the Gaussian kernels in radial basis function networks. In *European Symposium on Artificial Neural Networks – ESANN 2002*, pages 425–432, Bruges, 2002.
- [21] G.E.P. Box. Evolutionary Operation: A method for increasing industrial productivity. *Applied Statistics*, 6(2):81–101, June 1957.
- [22] G.E.P. Box and N.R. Draper. *Evolutionary Operation: A Statistical Method for Process Improvement*. Wiley, New York, 1969.
- [23] H.J. Bremermann. *Optimization through evolution and recombination*. Spartan Books, 1962.

BIBLIOGRAPHY

- [24] C.E. Brennen. *Cavitation and Bubble Dynamics*. Oxford Engineering Science Series. Oxford University Press, 1995.
- [25] P. Chaviaropoulos, V. Dedoussis, and K.D. Papailiou. On the 3D inverse potential target pressure problem. Part 1. Theoretical aspects and method formulation. *Journal of Fluid Mechanics*, 282:131–146, 1995.
- [26] A. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1):12–26, 1967.
- [27] C.A. Coello Coello. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. volume 1, pages 3–13, Washington, 1999.
- [28] C.A. Coello Coello. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41(2):113–127, 2000.
- [29] C.A. Coello Coello. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11–12):1245–1287, 2002.
- [30] C.A. Coello Coello, D.A. Van Veldhuizen, and G.B. Lamont. *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Publishers, 2002.
- [31] D. Corne, J. Knowles, and M. Oates. The Pareto envelope-based selection algorithm for multiobjective optimization. In *Parallel Problem Solving from Nature VI Conference – PPSN VI*, pages 839–848. Springer, 2000.
- [32] N.L. Cramer. A representation for the adaptive generation of simple sequential programs. In J.J. Grefenstette, editor, *International Conference on Genetic Algorithms and the Applications*, pages 183–187, 1985.
- [33] C. Darwin. *On the origin of species by means of natural selection*. Murray, London, 1859.
- [34] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2-4):311–338, 2000.
- [35] K. Deb and R.B. Agrawal. Simulated binary crossover for continuous search space. Technical report, IITK/ME/SMD-94027, Kanpur Genetic Algorithms Laboratory, 1994.

BIBLIOGRAPHY

- [36] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature – PPSN VI*, Paris, 2000.
- [37] V. Dedoussis, P. Chav iaropoulos, and K.D. Papailiou. On the 3D inverse potential target pressure problem. Part 2. Numerical aspects and application to duct design. *Journal of Fluid Mechanics*, 282:147–162, 1995.
- [38] B.H. Dennis, G. Dulikravich, and Z.-X. Han. Constrained shape optimization of airfoil cascades using a Navier–Stokes solver and a genetic/SQP algorithm. In *ASME IGTI' 99*, Indianapolis, 2002. ASME Paper 98-GT-441.
- [39] J.A. Désidéri and A. Janka. Hierarchical parametrization for multilevel evolutionary shape optimization with application to aerodynamics. In *EUROGEN 2003, Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems*, Barcelona, 2003.
- [40] D.J. Doorly and J. Peiró. Supervised parallel genetic algorithms in aerodynamic optimisation. In *13th AIAA Computational Fluid Dynamics Conference*, Snowmass Village, CO, 1997. AIAA-1997-1852.
- [41] A.F. Doukelis. *Study of the flowfield in the tip clearance region of compressor bladings*. PhD thesis, Cranfield University, 1998.
- [42] M. Drela and M.B. Giles. Viscous-inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, 25(10):1347–1355, 1987.
- [43] P. Drtina and M. Sallaberger. Hydraulic turbines—basic principles and state-of-the-art computational fluid dynamics applications. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 213(1):85–102, 1999.
- [44] D. Eby, R. C. Averill, W. F. Punch III, and E. D. Goodman. Evaluation of injection island GA performance on flywheel design optimization. In *3rd Conference on Adaptive Computing in Design and Manufacturing*, pages 121–136, Plymouth, 1998. Springer-Verlag.
- [45] M.A. El-Beltagy, P.B. Nair, and A.J. Keane. Metamodeling techniques for evolutionary optimization of computationally expensive problems: Promises and limitations. In *Genetic and Evolutionary Computation Conference – GECCO*, pages 196–203, San Francisco, 1999.
- [46] J. Elliott and J. Peraire. Aerodynamic design using unstructured meshes. AIAA Paper 96-1941, 1996.

BIBLIOGRAPHY

- [47] 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*, 10(4):421–439, 2006.
- [48] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. In *7th International Conference on Parallel Problem Solving from Nature — PPSN VII*, volume 2439 of *Lecture Notes in Computer Science*, pages 361–370. Springer-Verlag, 2002.
- [49] M. Emmerich and B. Naujoks. Metamodel-assisted multiobjective optimization with implicit constraints and its application in airfoil design. In *ERCOFTAC Design Optimization: Methods & Applications — International Conference & Advanced Course Program*, Athens, 2004.
- [50] I.K. Fodor. A survey of dimension reduction techniques. Technical report, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 2002.
- [51] L. J. Fogel. Autonomous automata. *Industrial Research Magazine*, 4(2):14–19, 1962.
- [52] L. J. Fogel. *On The Organization of Intellect*. PhD thesis, University of California, 1964.
- [53] L. J. Fogel, Owens A., and M. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, New York, 1966.
- [54] L.J. Fogel, P.J. Angeline, and D.B. Fogel. An evolutionary programming approach to self-adaptation on finite state machines. In *Proceedings of the 4th Annual Conference on Evolutionary Programming*, pages 355–365. MIT Press, 1995.
- [55] C.M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *5th International Conference on Genetic Algorithms*, pages 416–423, San Mateo, CA, 1993. Morgan Kaufmann Publishers.
- [56] N. Foster and G. Dulikravich. Three-dimensional aerodynamic shape optimisation using genetic and gradient search algorithms. *J. Spacecraft and Rockets*, 34(1), 1997.
- [57] R.M. Friedberg. A learning machine: Part I. *IBM J. Research and Development.*, 2(1):2–13, January 1958.

BIBLIOGRAPHY

- [58] R.M. Friedberg, B. Dunham, and J.H. North. A learning machine: Part II. *IBM J. Research and Development.*, 3(3):282–287, July 1959.
- [59] B. Fritzke. Fast learning with incremental RBF Networks. *Neural Processing Letters*, 1(1):2–5, 1994.
- [60] M. Galan, G. Winter, G. Montero, D. Greiner, J. Periaux, and B. Mantel. A transonic flow shape optimization using genetic algorithms. *ECCOMAS 96*, John Wiley & Sons, Paris, 1996.
- [61] D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155 – 170, 1983.
- [62] K. Georg, V. Christian, and N. Eberhard. Theory and application of axisymmetric endwall contouring for compressors. In *ASME Turbo Expo*, Vancouver, 2011. ASME Paper T2011-45624.
- [63] C. A. Georgopoulou. *Τεχνικές Βελτιστοποίησης για τον προγραμματισμό λειτουργίας Αεριο/Ατμοστροβιλικών μονάδων και το σχεδιασμό συνιστωσών τους*. PhD thesis, National Technical University of Athens, 2009.
- [64] C.A. Georgopoulou and K.C. Giannakoglou. A multi-objective metamodel-assisted memetic algorithm with strength-based local refinement. *Engineering Optimization*, 41(10):909–923, 2009.
- [65] C.A. Georgopoulou and K.C. Giannakoglou. *Multiobjective Metamodel-Assisted Memetic Algorithms*. Springer Series, 2009.
- [66] C.A. Georgopoulou, S.A. Kyriacou, K.C. Giannakoglou, P. Grafenberger, and E. Parkinson. Constrained multi-objective design optimization of hydraulic components using a hierarchical metamodel-assisted evolutionary algorithm. Part 1: Theory. In *24th IAHR, Symposium on Hydraulic Machinery and Systems*, Foz do Iguassu, October 2008.
- [67] T. Ghisu, T. Parks, D. Jaeggi, J. Jarrett, and P. Clarkson. The benefits of adaptive parametrization in multi-objective tabu search optimization. *Engineering Optimization*, 42(12):959–981, 2010.
- [68] K.C. Giannakoglou. Designing turbomachinery blades using evolutionary methods. In *ASME Turbo Expo '99*, Indianapolis, 1999. ASME Paper 99-GT-181.
- [69] K.C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, 38(1):43–76, 2002.

BIBLIOGRAPHY

- [70] K.C. Giannakoglou. The EASY (Evolutionary Algorithms SYstem) software, <http://velos0.ltt.mech.ntua.gr/EASY>, 2008.
- [71] K.C. Giannakoglou, A.P. Giotis, and M.K. Karakasis. Low-cost genetic optimization based on inexact pre-evaluations and the sensitivity analysis of design parameters. *Inverse Problems in Engineering*, 9:389–412, 2001.
- [72] K.C. Giannakoglou and I.C. Kampolis. *Multilevel Optimization Algorithms based on Metamodel- and Fitness Inheritance-Assisted Evolutionary Algorithms*. Springer-Verlag Series in Evolutionary Learning and Optimization, 2009.
- [73] K.C. Giannakoglou, I.C. Kampolis, P.I.K. Liakopoulos, M.K. Karakasis, D.I. Papadimitriou, V.G. Asouti, and T. Zervogiannis. Aerodynamic shape optimization methods on multiprocessor platforms. *ParCFD 2007*, Antalya, May 21-24, 2007.
- [74] A.P. Giotis. *Χρήση Εξελικτικών τεχνικών, υπολογιστικής ευφυίας και μεθόδων υπολογιστικής ρευστομηχανικής στη βελτιστοποίηση - αντίστροφη σχεδίαση πτερυγώσεων στροβιλομηχανών, μέσω παράλληλης επεξεργασίας*. PhD thesis, National Technical University of Athens, 2003.
- [75] A.P. Giotis and K.C. Giannakoglou. Single- and multi-objective airfoil design using genetic algorithms and artificial intelligence. *EUROGEN 99, Evolutionary Algorithms in Engineering and Computer Science*, Miettinen K. et al (Eds.), John Wiley & Sons, Jyvaskyla, 1999.
- [76] D.E. Goldberg. *Genetic Algorithms in search, optimization & machine learning*. Addison-Wesley, 1989.
- [77] D.E. Goldberg and K. Deb. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In *Foundations of Genetic Algorithms*, pages 69–93, 1991.
- [78] P. Grafenberger, E. Parkinson, C.A. Georgopoulou, S.A. Kyriacou, and K.C. Giannakoglou. Optimization of hydraulic machinery by exploiting previous successful designs. In *25th IAHR, Symposium on Hydraulic Machinery and Systems*, Timisoara, September 2010.
- [79] N. Hansen, S.D. Muller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

BIBLIOGRAPHY

- [80] N. Hansen and A. Ostermeier. Convergence properties of evolution strategies with the derandomized covariance matrix adaptation: The $(\mu/\mu_i, \lambda)$ -ES. *EUFIT'97*, pages 650–654, 1997.
- [81] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [82] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, New Jersey, 2nd edition, 1999.
- [83] F. Herrera, M. Lozano, and C. Moraga. Hierarchical distributed genetic algorithms. *International Journal of Intelligent Systems*, 14(9):1099–1121, 1999.
- [84] J. Holland. In *Adaptation In Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975.
- [85] J.H. Holland. Outline for a logical theory of adaptive systems. *J. ACM*, 9(3):297–314, July 1962.
- [86] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. In *First IEEE Conference on Evolutionary Computation, 1994*, volume 1, pages 82–87 vol.1, 1994.
- [87] A. Jameson. Optimum aerodynamic design using CFD and control theory. AIAA Paper 95-1729-CP, 1995.
- [88] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computation*, 3:33–260, 1988.
- [89] A. Jameson and J. Reuther. Control theory based airfoil design using the Euler equations. AIAA Paper 94-4272-CP, 1994.
- [90] Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5):481–494, October 2002.
- [91] I.T. Jolliffe. In *Principal Component Analysis, Springer Series in Statistics, 2nd ed.* Springer, N.Y., 2002.
- [92] I. C. Kampolis. *Πολυεπίπεδοι, Πολυεπεξεργαστικοί Αλγόριθμοι Αεροδυναμικής Βελτιστοποίησης στις Στροβιλομηχανές*. PhD thesis, National Technical University of Athens, 2009.

BIBLIOGRAPHY

- [93] I.C. Kampolis and K.C. Giannakoglou. A multilevel approach to single- and multiobjective aerodynamic optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33-40):2963–2975, 2008.
- [94] I.C. Kampolis and K.C. Giannakoglou. Distributed evolutionary algorithms with hierarchical evaluation. *Engineering Optimization*, 41(11):1037–1049, 2009.
- [95] I.C. Kampolis and K.C. Giannakoglou. Synergetic use of different evaluation, parameterization and search tools within a multilevel optimization platform. *Applied Soft Computing*, 11(1):645–651, 2011.
- [96] I.C. Kampolis, D.I. Papadimitriou, and K.C. Giannakoglou. Evolutionary optimization using a new radial basis function network and the adjoint formulation. *Inverse Problems in Science and Engineering*, 14(4):397–410, 2006.
- [97] I.C. Kampolis, A.S. Zymaris, V.G. Asouti, and K.C. Giannakoglou. Multilevel optimization strategies based on metamodel-assisted evolutionary algorithms, for computationally expensive problems. In *2007 Congress on Evolutionary Computation – CEC '07*, Singapore, September 2007.
- [98] M.K. Karakasis. *Ιεραρχικοί, κατανεμημένοι εξελικτικοί αλγόριθμοι και υπολογιστική ευφυία στη σχεδίαση βέλτιστων αεροδυναμικών μορφών, σε περιβάλλον πολυεπεξεργασίας*. PhD thesis, National Technical University of Athens, 2006.
- [99] M.K. Karakasis and K.C. Giannakoglou. On the use of metamodel-assisted, multi-objective evolutionary algorithms. *Engineering Optimization*, 38(8):941–957, 2006.
- [100] M.K. Karakasis, A.P. Giotis, and K.C. Giannakoglou. Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization. *International Journal for Numerical Methods in Fluids*, 43(10-11):1149–1166, 2003.
- [101] M.K. Karakasis, D.G. Koubogiannis, and K.C. Giannakoglou. Hierarchical distributed evolutionary algorithms in shape optimization. *International Journal for Numerical Methods in Fluids*, 53(3):455–469, 2007.
- [102] N.B. Karayiannis and G.W. Mi. Growing radial basis neural networks: Merging supervised and unsupervised learning with network growth techniques. *IEEE Transactions on Neural Networks*, 8(6):1492–1506, 1997.

BIBLIOGRAPHY

- [103] A. Keane and B. Nair. *Computational Approaches to Aerospace Design*. John Wiley & Sons, Chichester UK, 2nd edition, 2005.
- [104] P. Kevin, G.T. Mark, S. Kiran, and D. Soumitr. Optimization of a 3-stage booster Part 1: the axisymmetric multi-disciplinary approach to compressor design. In *ASME Turbo Expo*, Vancouver, 2011. ASME Paper GT2011-46569.
- [105] M. Kiemele, S. Schmidt, and R. Berdine. *Basic Statistics, tools for continuous improvement*. Air Academy Press & Associates, Colorado Springs, 1996.
- [106] R.T. Knapp, J.W. Daily, and F.G. Hammitt. *Cavitation*. Engineering Societies Monographs. McGraw-Hill, 1970.
- [107] J. Knowles and D. Corne. The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation. In *1999 Congress on Evolutionary Computation – CEC 1999*, volume 1, pages 98–105. IEEE Press, 1999.
- [108] J. Knowles and D. Corne. M-PAES: A memetic algorithm for multiobjective optimization. In *2000 Congress on Evolutionary Computation – CEC 2000*, pages 325–332. IEEE Press, 2000.
- [109] L. Koch. Flächenrückführung für Turbinenschaufeln. Diploma thesis, Johannes Kepler University Linz, 2004.
- [110] J. Kolodner. Maintaining organization in a dynamic long-term memory. *Cognitive Science*, 7(iv), pages 243–80, 1983.
- [111] J. Kolodner. Reconstructive memory: A computer model. *Cognitive Science*, 7(iv), pages 281–28, 1983.
- [112] J. Kolodner. Arguing by Analogy in Law: A Case-Based Model. in D.H. Helman (Ed.), *Analogical Reasoning: Perspectives of Artificial Intelligence. Cognitive Science, and Philosophy*, 1988.
- [113] J. Kolodner. Improving human decision making through case-based decision aiding. *AI Magazine*, 12(2):52–68, 1991.
- [114] J. Kolodner. In *Case-Based Reasoning*. Morgan Kaufmann Publication, San Mateo, CA, 1993.

BIBLIOGRAPHY

- [115] E.A. Kontoleontos. *Συμβολή στο σχεδιασμό θερμο-ρευστοδυναμικών διατάξεων με χρήση αιτιοκρατικών και εξελικτικών μεθόδων βελτιστοποίησης*. PhD thesis, National Technical University of Athens, 2012.
- [116] E.A. Kontoleontos, V.G. Asouti, and K.C. Giannakoglou. An asynchronous metamodel-assisted memetic algorithm for CFD-based shape optimization. *Engineering Optimization*, 44(2):157–173, 2011.
- [117] J. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, 1994.
- [118] N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE Transactions on Evolutionary Computation*, 9(5):474–488, 2005.
- [119] J. Le Foll. A theory of representation of the properties of boundary-layers on a plane. In *Seminar on Advanced Problems in Turbomachinery*. VKI, Rhodes–Saint Genése, 1965.
- [120] J. Lee and P. Hajela. Parallel genetic algorithm implementation in multi-disciplinary rotor blade design. *J. Aircraft*, 33(5):962–969, 1996.
- [121] P.I.K. Liakopoulos. *Γένεση μηδομημένων πλεγμάτων και διαχείρισή τους σε μεθόδους ανάλυσης και βελτιστοποίησης συνιστωσών στροβιλομηχανών και εφαρμογές, αξιοποιώντας τεχνολογίες πλέγματος (Grid Computing)*. PhD thesis, National Technical University of Athens, 2008.
- [122] D. Lim, Y.-S. Ong, Y. Jin, B. Sendhoff, and B.-S. Lee. Efficient hierarchical parallel genetic algorithms using grid computing. *Future Generation Computer Systems*, 23(4):658–670, 2007.
- [123] A. Marcel and S. Ulrich. High-dimensional constrained multiobjective optimization of a fan stage. In *ASME Turbo Expo*, Vancouver, 2011. ASME Paper GT2011-45618.
- [124] T.J. Martin and G. S. Dulikravich. Aero-thermal analysis and optimization of internally cooled turbine airfoils. ISABE 97-7165, 1997.
- [125] K. Mathioudakis, K.D. Papailiou, N. Neris, C. Bonhommet, G. Albrand, and U. Wenger. An annular cascade facility for studying tip clearance effects in high speed flows. In *XIII ISABE Conference*, Chattanooga, Tennessee, 1997.
- [126] Z. Michalewicz and D.B. Fogel. *How to solve it: Modern Heuristics*. Springer-Verlag, 2000.

BIBLIOGRAPHY

- [127] Z. Michalewicz and M. Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4:1–32, 1996.
- [128] Z. Michalewicz and N. Attia. Evolutionary optimization of constrained problems. In *3rd Annual Conference on Evolutionary Programming*, pages 98–109. World Scientific Publishing, 1994.
- [129] K. Miettinen. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, Boston, 1999.
- [130] C. Moore, M. Lehane, and C. Proce. Case-based reasoning for decision support in engineering design. Proc. IEE Colloquium on Case-Based Reasoning: Prospects for Applications, Digest No: 1994/057, 1994.
- [131] A.K. Morales and C.V. Quezada. A universal eclectic genetic algorithm for constrained optimization. In *6th European Congress on Intelligent Techniques & Soft Computing, EUFIT'98*, pages 518–522. Verlag Mainz, 1998.
- [132] A. Nebro, J. Durillo, F. Luna, B. Dorronsoro, and E. Alba. MOCell: A cellular genetic algorithm for multiobjective optimization. *Int. J. Intell. Syst.*, 24(7):726–746, 2009.
- [133] Y.S. Ong and A.J. Keane. Meta-lamarckian learning in memetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2):99–110, 2004.
- [134] Y.S. Ong, M.H. Lim, N. Zhu, and K.W. Wong. Classification of adaptive memetic algorithms: a comparative study. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 36(1):141–152, 2006.
- [135] D.I. Papadimitriou. *Συζυγείς (Adjoint) διατυπώσεις για την ανάλυση - σχέδιαση πτερυγώσεων στροβιλομηχανών και τη βέλτιστη προσαρμογή πλέγματος με a posteriori εκτίμηση σφάλματος*. PhD thesis, National Technical University of Athens, 2006.
- [136] D.I. Papadimitriou and K.C. Giannakoglou. Total pressure loss minimization in turbomachinery cascades using a new continuous adjoint formulation. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 221(6):865–872, 2007.
- [137] K.D. Papailiou. Optimization d’ aubage de compresseur à forte charge sur la base des théories de couches limites. Thèse de Doctorat en Sciences Appliquées, University de Liège, 1969.

BIBLIOGRAPHY

- [138] K.D. Papailiou. Le Foll's method and the calculation of attached and separated two-dimensional boundary layers. In *Separated Flows in Turbomachinery Components*, volume 2. VKI, Rhodes–Saint Genése, 1981.
- [139] K.D. Papailiou, F. Nurzia, and A. Satta. The inclusion of surface curvature effects in the calculation of the turbulent boundary layer with integral methods. Internal Note 37, VKI, 1970.
- [140] E.D. Papantonis. *Μικρά Υδροηλεκτρικά Έργα*. Συμεών, Athens, 2008.
- [141] D. W. Peaceman and H. H. Rachford. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.*, 3:28–41, 1955.
- [142] J. Périaux, M. Sefrioui, B. Stoufflet, B. Mantel, and E. Laporte. Robust genetic algorithms for optimization problems in aerodynamic design. In *Genetic Algorithms in Engineering and Computer Science*, pages 371–369. John Wiley & Sons, 1995.
- [143] O. Pironneau. *Optimal shape design for elliptic systems*. Springer-Verlag, New York, 1984.
- [144] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.
- [145] E.S. Politis, K.C. Giannakoglou, and K.D. Papailiou. High-speed flow in an annular cascade with tip clearance: Numerical investigation. In *ASME Gas Turbine & Aeroengine Congress and Exhibition*, Stockholm, 1998. ASME Paper 98-GT-247.
- [146] R. Popa, F. Popa, Popa B., and D Zachia-Zlatea. Optimization of the weekly operation of a multipurpose hydroelectric development, including a pumped storage plant. In *25th IAHR, Symposium on Hydraulic Machinery and Systems*, Timisoara, September 2010.
- [147] D. Powell and M.M. Skolnick. Using genetic algorithms in engineering design optimization with non-linear constraints. In *5th International Conference on Genetic Algorithms*, pages 424–431. Morgan Kaufmann Publishers Inc., 1993.
- [148] D. Quagliarella and A.D. Cioppa. Genetic algorithms applied to the aero-dynamic design of transonic airfoils. *J. Aircraft*, 32(4):889–891, 1995.

BIBLIOGRAPHY

- [149] R.G. Raimunda da Silva, E. Ramirez Camacho and N Manzanares Filho. Global optimization based on metamodel construction app lied to design axial turbomachinery cascades using cfd. In *25th IAHR, Symposium on Hydraulic Machinery and Systems*, Timisoara, September 2010.
- [150] A. Ratle. Accelerating the convergence of evolutionary algorithms by fitness landscape approximation. *PPSN V, Lecture Notes in Computer Science* 1498, pp. 87-96, Springer-Verlag, 1998.
- [151] I. Rechenberg. Cybernetic solution path of an experimental problem. *Library Translation 1122, Royal Aircraft Establishment, Farnborough*, 1965.
- [152] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. PhD thesis, Technical University of Berlin, 1971.
- [153] C. Riesbeck and R. Schank. In *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Cambridge, MA, 1989.
- [154] E. Rissland and D. Skala. Combining case-based and rule-based reasoning: A heuristic approach. 11th International Joint Conference on Artificial Intelligence, IJCAI-89: pp. 524-30. Detroit, Michigan, 1989.
- [155] P. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.
- [156] R. Roy and A. Tiwari. Generalised regression GA for handling inseparable function integration: Algorithm and application. *Parallel Problem Solving from Nature - PPSN VII, Lecture Notes in Computer Science*, 2439:452–461, 2002.
- [157] R. Roy and A. Tiwari. Variable dependence interaction and multi-objective optimizations. *Genetic and Evolutionary Computation Conference, New York*, 2002.
- [158] R. Salomon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39:263–278, 1996.
- [159] J.D. Schaffer and L.J. Eshelman. On crossover as an evolutionary viable strategy. *4th International Conference on Genetic Algorithms, San Diego*, pages 61–68, 1991.

BIBLIOGRAPHY

- [160] J.D. Schaffer and A. Morishima. An adaptive crossover distribution mechanism for genetic algorithms. *Second International Conference on Genetic Algorithms*, pages 36–40, 1987.
- [161] R. Schank. In *Dynamic memory: a theory of reminding and learning in computers and people*. Cambridge University Press, Cambridge, 1982.
- [162] R. Schank and R. Abelson. In *Scripts, Plans, Goals and Understanding*. Erlbaum, Hillsdale, New Jersey, 1977.
- [163] R. Schmidl. *Contribution to the cavitation optimization of Kaplan blades by numerical flow simulation and evolutionary strategies*. PhD thesis, Graz University of Technology, 2007.
- [164] H. P. Schwefel. *Evolutionsstrategie und numerische Optimierung*. PhD thesis, Technical University of Berlin, 1975.
- [165] M. Sefrioui. *Algorithmes Evolutionnaires pour le Calcul Scientifique. Application à l'electromagnétisme et à la Mécanique des Fluides Numériques*. PhD. thesis, University of Paris 6, 1998.
- [166] M. Sefrioui and J. Péraux. A hierarchical genetic algorithm using multiple models for optimization. In *6th International Conference on Parallel Problem Solving from Nature – PPSN VI*, volume 1917 of *Lecture Notes in Computer Science*, pages 879–888, Paris, 2000. Springer.
- [167] S. Slade. Case-based reasoning: A research paradigm. *AI Magazine*, 12(1):42–55, 1991.
- [168] N. Smith and D. Schorr. Fact retrieval and the paradox of interference. pages 438–464. *Cognitive Psychology*, 10, 1978.
- [169] W.M. Spears. Crossover or mutation? In *Foundations of Genetic Algorithms 2*, pages 221–237. Morgan Kaufmann, 1992.
- [170] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1995.
- [171] P.D. Surry and N.J. Radcliffe. The COMOGA method: Constrained optimisation by multiobjective genetic algorithms. *Control and Cybernetics*, 26:391–412, 1997.

BIBLIOGRAPHY

- [172] P.D. Surry, N.J. Radcliffe, and I.D. Boyd. A multi-objective approach to constrained optimisation of gas supply networks : The COMOGA method. In *Evolutionary Computing. AISB Workshop. Selected Papers, Lecture Notes in Computer Science*, pages 166–180. Springer-Verlag, 1995.
- [173] IEC TC/SC. *CEI/IEC 60193: Hydraulic turbines, storage pumps and pump-turbines – Model acceptance tests*. ANSI, 1999.
- [174] A. Thiruvengadam. *Handbook of cavitation erosion*. Hydronautics, inc., 1974.
- [175] A.N. Tikhonov and V. Arsénine. *Méthodes de Résolution de problèmes mal posés*. Editions MIR, Moscou, 1976.
- [176] A.N. Tikhonov, A.V. Goncharsky, V.V. Stepanov, and A.G. Yagola. *Numerical Methods for the Solution of Ill-Posed Problems*. Kluwer Academic Publishers, 1995.
- [177] E. Tulving. Episodic and semantic memory. in, organisation of memory. pages 381–403. Academic Press, 1977.
- [178] E. Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. *Journal of Computational Physics*, 72:277–298, 1987.
- [179] G.D. van Albada, B. van Leer, and W.W. Roberts. A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics*, 108:76–84, 1982.
- [180] B. van Leer. Towards the ultimate conservative difference scheme, V. A second-order sequel to Godunov’s method. *Computational Physics*, 32:101–136, 1979.
- [181] I. Watson and S. Abdullah. Developing case-based reasoning systems: A case study in diagnosing building defects. pages 1/1–1/3. Proc. IEE Colloquium on Case-Based Reasoning: Prospects for Applications, Digest No: 1994/057, 1994.
- [182] S. Weissenberger. Effiziente numerische Simulation der inkompressiblen Euler-Gleichungen mit Anwendungen im Turbinenbau. Diploma thesis, Johannes Kepler University Linz, 2005.
- [183] L. Wittgentstein. *Philosophical investigations*. Blackwell, 1953.

BIBLIOGRAPHY

- [184] T. Zervogiannis, D.I. Papadimitriou, and K.C. Giannakoglou. Total pressure losses minimization in turbomachinery cascades using the exact Hessian. *Computer Methods in Applied Mechanics and Engineering*, 199(41-44):2697–2708, 2010.
- [185] E. Zitzler, D. Brockhoff, and L. Thiele. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In *Conference on Evolutionary Multi-Criterion Optimization – EMO*, 2007.
- [186] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8:173–195, June 2000.
- [187] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *EUROGEN 2001, Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems*, Barcelona, 2002. CIMNE.
- [188] E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The strength Pareto approach. TIK-Report No. 43, Computer Engineering and Communication Networks Lab, ETH Zurich, 1998.